



GENERAL AUTOMATION, INC.

1055 So. East Street, Anaheim, Calif. 92805

H

T. NELSON , SYS CONSLTNT
BOX 3

SCHOOLEYS MTN, NEW JERSEY 070

SPC-16 Industrial Automation Computer



GENERAL AUTOMATION, INC.

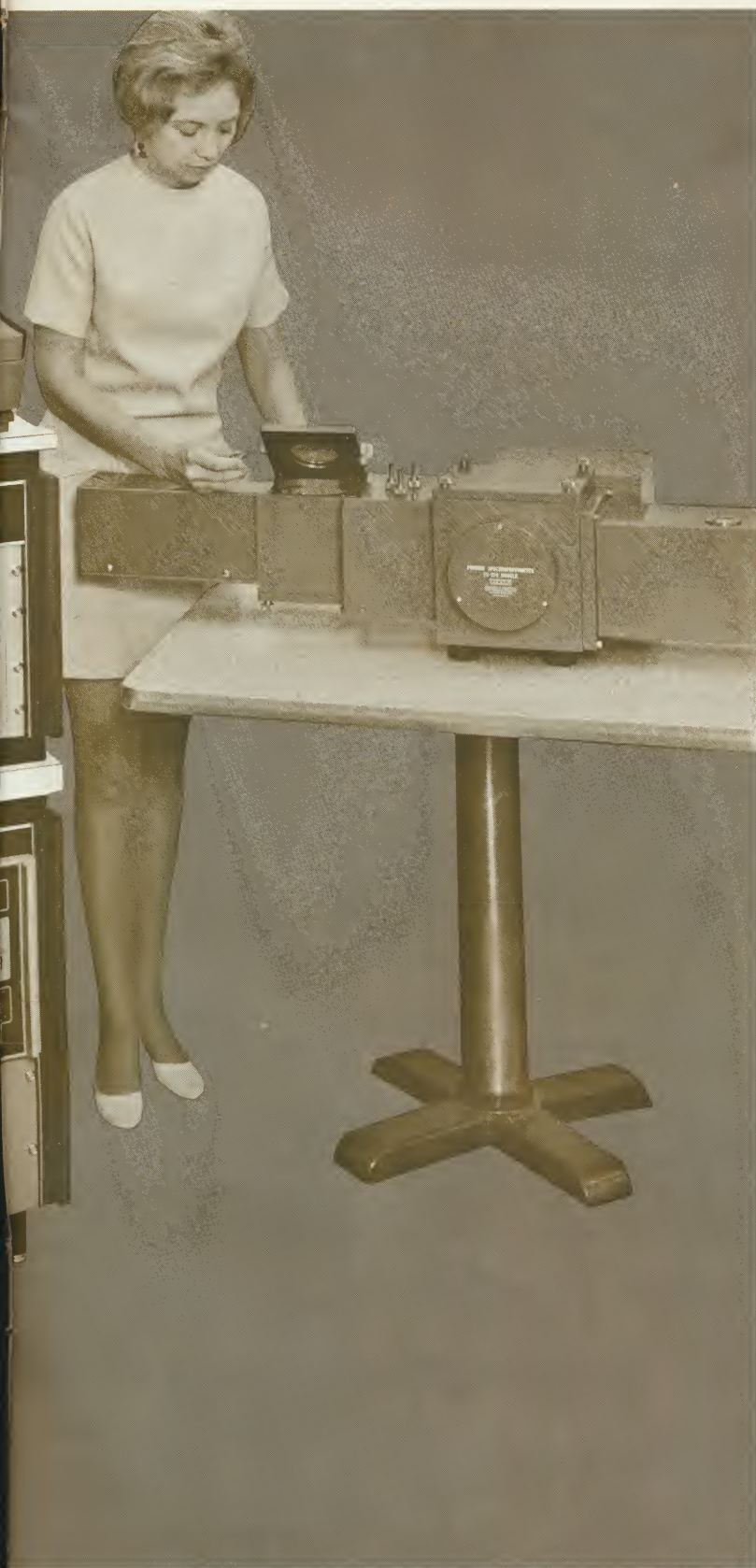
CONTENTS

	PAGE
INTRODUCTION	1
RELIABILITY	3
INDUSTRIAL APPLICATIONS	4
SOFTWARE	6
SYSTEM CONSOLE	10
COMPUTER POWER	11
ROM WITH SPC-16	11
ORGANIZATION	12
ADDRESSING	12
INSTRUCTION REPERTOIRE	14
INSTRUCTION CLASSES	14
SYSTEM I/O	16
SYSTEM SUMMARY	17
SPECIFICATIONS	18
MINI-CONTROLLERS	19
DATA PROCESSING	
PERIPHERAL UNITS	20
THE TOTAL-SOLUTION COMPANY	21

INTRODUCTION

General Automation's SPC-16 is the "super-performance" 16-bit computer in GA's compatible family of fourth-generation industrial automation equipment. It solves high-performance automation requirements with maximum reliability, dependability and economy. The SPC-16 automation computer blends an unmatched combination of fast response time and powerful computing capability with fail safe features and industrial reliability. In addition, the SPC-16 performs like two computers in one; hardware features accommodate real-time dedicated control applications in the foreground, while performing background batch processing. These features — backed by a full range of software packages, technical services and maintenance support — make the SPC-16 the ideal computer system for total automation solutions in the industrial environment. In addition, its versatile, general purpose capabilities make it valuable in process control, communications, laboratory, medical, and information processing applications.





DESIGNED FOR TOTAL MANUFACTURING AUTOMATION

Fast, powerful and versatile, the economical SPC-16 is ideally suited for computer-based automation. A fully automated manufacturing or processing program plays a vital part in helping you to achieve production efficiency, operating economy, and improved service to your customers with lower priced, better quality products. The SPC-16 also enables you to optimize your company's resources in research, engineering, manufacturing, quality control, and information processing. More importantly, the SPC-16 can increase your productivity and profits with a minimum initial investment and fast payback. With GA's product compatibility, applications experience, and specialized technical services, GA makes it easy and practical to automate.

OPTIMIZING COMPUTER USE AND COST

The SPC-16 can be used independently or combined into a total system. It can be easily added to an existing system with a minimum of disruption and costs. It can also be used as today's basic automation system, while assuring easy addition of expansion or supervisory/management equipment as future needs arise. The SPC-16 is part of GA's growing family of compatible computer products in GA's Distributed Computer Control. This concept provides a more functional and efficient method of manufacturing automation; it matches the size and capability of the computing system with the size and complexity of the application. Thus, each computer is priced commensurately with the level of work performed, offering maximum operating efficiency and profit returns with minimum expansion costs as well as minimum initial investment.

HARDWARE FOREGROUND/BACKGROUND PROCESSING

With its dual capability of foreground/background processing, the SPC-16 computer system can perform an unlimited variety of automation solutions and computing functions in the manufacturing environment. Hardware features in the SPC-16 enable the computer to perform foreground on-line dedicated-control processing as well as background batch processing. Foreground processing includes manufacturing and process control, quality control and testing, materials handling and inventory control. Background batch processing includes program generation and fast-response, low-overhead interrupt processing.

TOP PERFORMANCE IN A SMALL PACKAGE

The new SPC-16 has the kind of performance and reliability you would expect from much larger — and much more costly — 16-bit machines. The high-speed core memory of this general purpose digital computer is 960 nanoseconds fast, and is field expandable to 32K in 4K increments. The 480 nanosecond ROM is field expandable in 512, 1024, or 2048 word increments. The SPC-16 also has a powerful repertoire of bit/byte/word instructions; it saves memory capacity and access time by performing logical and arithmetic operations in 16 general purpose accumulator/index registers. The new computer incorporates a unique visual debug aid, "Autovue," that reduces programming time, optimizes program writing, serves as a dynamic diagnostic aid, and assists in computer installation debugging. All of these features, and many more, make the SPC-16 the fastest, most powerful and most versatile of any contemporary 16-bit computer system.

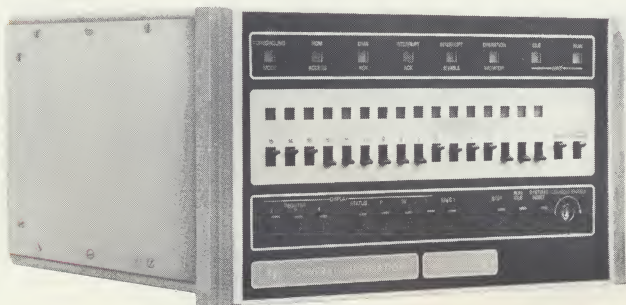
High Speed, High Through-put Versatile I/O System — 520K words per second programmable I/O
☐ 1.04 million words per second data channels with arithmetic and logical functions
☐ Unlimited hardware priority interrupts for fast response time (960 nanoseconds) and low overhead.

Flexible Computer Organization — Hardware foreground/background processing
☐ 16 programmable hardware registers
☐ Micro control bus.

Versatile Addressing Techniques — Direct
☐ Indirect
☐ Program relative
☐ Base relative
☐ Literal
☐ Complete complement of indexing.

Fast Modular Memory Expandable to 32K of 16-bit Words — Core: 960 nanoseconds
☐ Read only memory (ROM): 480 nanoseconds.

Effective Use of ROM — Versatile execute instruction
☐ Base and program relative addressing
☐ Sixteen general purpose registers
☐ Fast ROM instruction execution time
☐ Free ROM-R/W memory transfer
☐ Completely interchangeable memory modules
☐ Relocatable power-fail/auto-restart interrupt vectors.



SYSTEM FLEXIBILITY AND INTERFACING

Mini-controllers are pre-engineered GA system interface modules that provide modularity and versatility. The functional, plug-in design eliminates excessive or redundant electronics and permits economical system configuring, servicing, and expansion. Over 70 Mini-controllers are available off-the-shelf, providing system flexibility in interfacing with mechanisms, devices, instruments, sensors, and peripherals. Peripherals supplied by GA include printers, keyboards, teletypewriters, card/paper tape reader and punches, mag tapes, plotters, rotating memory and disk storages, and data sets.

PROVEN PROGRAMMING SUPPORT

Every SPC-16 includes support software packages to assist system integration and to facilitate project completion. In addition, a wide selection of optional service-oriented and functional application "Automate" programs are available from regional GA Technical Application Centers.

FULL RANGE OF AUTOMATION SERVICES

As the "total solution" automation company, GA supports its entire hardware/software product line with a full range of technological services for successfully implementing your automation project. GA's staff of experienced, highly qualified automation experts provide professional aid for any phase of project analysis or implementation to complete turn-key project responsibility. These services include:
☐ Applications analysis and consultation,
☐ Systems engineering,
☐ Project documentation,
☐ Programming, and
☐ User training. Computer warranty and installation/maintenance support are also available with each SPC-16 system.

THE "TODAY/TOMORROW" AUTOMATION SYSTEM

We call the SPC-16 the "today/tomorrow" automation computer. With the SPC-16 you receive — in addition to the best industrial automation system for today's manufacturing production needs — the basis system for tomorrow's ever-growing, ever-more-complex automation requirements. The growth potential of such a system is unlimited. The comprehensive GA product line, its hardware/software compatibility, the wide variety of GA interface units, and extensive applications experience make expansion with GA simple, efficient, and economical. Your local GA representative will gladly give you the facts on how the "super performance" SPC-16 system can help you keep pace with your industry's growth . . . to help you maintain the competitive edge . . . and to provide you with increased productivity and higher profits.

RELIABILITY

FOURTH GENERATION TECHNOLOGY AND RELIABILITY

The high performance and long-life reliability of the SPC-16 is the result of GA's unique fourth-generation design, materials, and manufacturing and testing techniques. Fourth generation technology used in all GA computer products, begins at the critical design state, is supported by fourth-generation construction techniques and built-in system safeguards, and is assured with exhaustive component/system testing and checkout. These standards of quality make the rugged SPC-16 the ideal computer system for automated operation in the industrial environment.

EQUIPMENT RELIABILITY

Fourth generation reliability begins at the critical design state with stringent "worst-case" criteria. Mass production, state-of-the-art techniques utilize small, medium and large scale integrated circuits with completely wire-free construction. Large-board technology are combined with sturdy materials such as low-heat integrated circuits and wide-temperature lithium ferrite memory modules to provide a solid basis for reliable operation. All parts are manufactured for complete interchangeability and compatibility, and the system is enclosed with GA's exclusive environment-reliable packaging.

All SPC-16 components and subsystems are 100% inspected and tested before assembly. Integrated circuits, for example, are inspected after 48 hours in a 125°C temperature oven. Upon completion, each system is cycle-sequence tested for seven days in an environmental chamber ranging in temperature from 0°C to 50°C.

Such exhaustive testing eliminates defective components or those which have infant-mortality tendencies; this assures high-performance operation of the system in even worst-case noise/temperature environmental conditions.

FAIL-SAFE GROUP

A special group of fail-safe features is an integral part of the SPC-16. This group protects both system operation and user processes from the costly consequence of power interruptions and transients, component breakdown, system shutdown, or programming bugs.

Power Fail Detection/Shutdown and Restart:

In addition to normal detection, shutdown, and restart functions, this feature generates a signal on the System Safe Line at the advent of a power failure.

Interrupts: For full system protection, the Power-Fail/Auto-Restart Interrupts cannot be inhibited.

Cold Start: Relocatable Interrupt Vectors allow ROM to protect a loader program; the computer can therefore automatically reload the main program after a power failure or other malfunction.

Operations Monitor Alarm:

The Operations Monitor Alarm (OHA) assures proper program execution. A failure condition of the Operations Monitor Alarm is indicated on the system console and generates a signal on the System Safe Line.

MAINTAINABILITY

Each section of the SPC-16 is interchangeable, even to memory stacks with loaded programs, and is constructed for on-site replacement capability. Modules have no slot sensitivity, and large-board, pre-engineered techniques simplify maintenance. Trained technicians, available from major locations throughout the U.S. and Europe, simply replace your board or memory in the field, send the original to the factory, then replace the temporary unit with the repaired module. Test and verify programs supplied with the SPC-16 pinpoint faults for quick troubleshooting, and margin test switches are included on every power supply.



INDUSTRIAL APPLICATIONS

General Automation has successfully completed hundreds of automation projects for both individual and multiple-system installations. Shown below are some typical industrial automation applications where GA has applied its products and services to increase productivity, lower operating costs, improve profits and produce better quality products.

MANUFACTURING AUTOMATION

General Automation computers are designed to provide reliable automation solutions in the manufacturing environment. For example, a GA computer is the central control element of a three axis machine tool. The computer generates the sequence of control signals required to move the machine on its three axes as well as monitoring the machine's actual position to ensure the system's accuracy. In addition, displays showing both the system's parameters and any error conditions in the machine tool are available on demand.

In the full system, one GA computer controls a number of machine tools independently. Any of these several patterns stored in memory can be selected to run on any of the machines simultaneously.

PROCESS DATA ACQUISITION AND CONTROL

General Automation computers are field-proven in hundreds of data acquisition/process control applications. GA computer-based automation systems provide the basic process monitoring functions of analog and digital input scanning; conversion to engineering units of flow, temperature, pressure, etc.

High-speed and low-level analog input signals are scanned and converted. The converted, filtered analog value is then compared against process high/low limits and set appropriate alarm flags or a return-to-normal flag. All of these limits, flags, and control points are stored in the computer's core memory and, because they are variable, they can be set and reset by the operator as required.



INSTRUMENTATION AND LABORATORY AUTOMATION

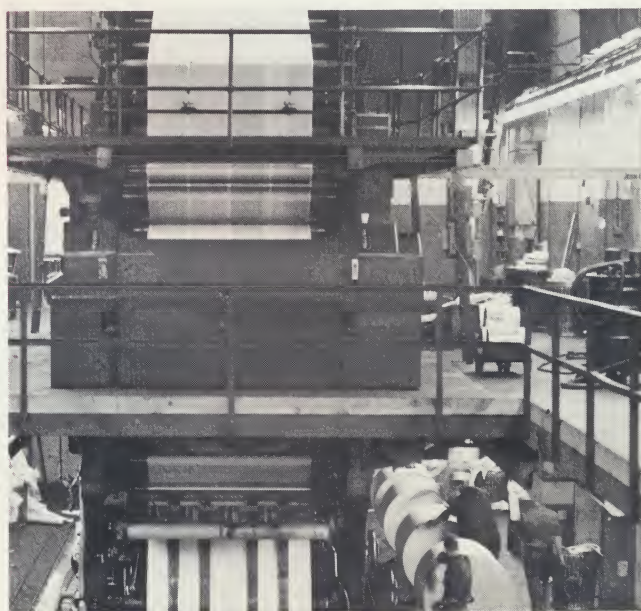
The growing demand for analytical data, faster data acquisition and reduction, and real time/on-line control in the laboratory, medical, and quality control environments dictate a computer-based automation solution. GA has supplied successful solutions to a wide variety of industries, including the petroleum, chemical, pharmaceutical, instrument manufacturing, medical research, and environment control for air and water pollution.

Other typical GA laboratory automation applications include mass spectrometers, optical emission spectrometers, densitometers, electron microprobe analyzers, and analytical instruments using slow-speed analog readout devices such as strip chart recorders and X-Y plotters.

COMMUNICATIONS

Current growth pattern indicates computer control systems have only touched the surface of potential communications applications. An example is the synchronous exchange of information on medium to high speed data links.

In this application, involving multi-processor or multi-device complexes, remote computers perform process control or data gathering/distribution tasks, and provide the central computer with information and data for process and/or management decisions and commands. For example, in the petrochemical industry, GA computers located over a wide area perform process control functions (controlling fuel mixtures) and report management information (fuel volumes) to a central management information system. Other uses include data concentration with the addition of binary synchronous communication capabilities.



APPLICATION	SYSTEM FUNCTION
COMMUNICATIONS	
TELEPHONE/TELEX:	Controls switching for telephone and telex systems.
LABORATORY	
MEDICAL:	On-line digitizing and recording of human EKG data. Acquisition and control of amino-acid analysis data.
SCIENTIFIC:	Processing and conversion of narrowband digital samples as part of spectral-analysis system. Preprocesses data for large-scale computer, acts as analyzer/computer interface. Data acquisition and control of gas chromatographs and mass spectrometers.
ELECTRONICS	
PERIPHERALS:	Computer interfacing and multi-unit control of I/O peripherals for large-scale computer installation. I/O units include line printer, card reader, card punch, and magnetic tapes.
PRODUCTION:	Automatic development, formatting, and conversion of instruction programs for numerically-controlled PCB drilling machines.
TEST AND CHECKOUT:	Checkout, calibration, and correlation of non-linear transducer test data, generating calibration curves via pre-set points and outputting data in volts.
MANUFACTURING	
PROCESS PLANT:	Automation of entire pilot plant engaged in metal-extraction processes.
PRODUCTION:	Stored-program automation of reed-relay switch coil-winding machines.
TESTING:	Control, monitoring, and recording of gasoline/diesel-engine test-cell events.
TOOLING:	Data processing, interpolation, and on-line/real-time control of numerically-controlled machine tools.
MATERIAL HANDLING:	Automated material handling and inventory control system.
PRINTING PRESS:	Monitoring and control of printing presses, including maintenance of pre-set inking.
TYPESETTING:	Automation of formatting and typesetting in high-volume newspaper operations.



APPLICATION	SYSTEM FUNCTION
POWER	
MONITORING:	Monitors power obtained from upstream power plant.
AEROSPACE	
PRODUCTION:	Automatic riveting machines for aircraft wings.
TESTING:	Fatigue testing and monitoring.
AUTOMOTIVE	
TESTING:	Control, monitoring, and recording of gasoline/diesel-engine test-cell events. Production testing of carburetors.
TRANSPORTATION	
RAILROAD:	Car counting and scale weighing system.
AUTOMOBILE:	Centralized computer control of an electronic traffic control system.
AIRLINE:	Monitors and displays airline flight arrivals and departures.
PETROLEUM	
OIL FIELD:	On-site acquisition and processing data received from drilling rigs on depth, density, etc. Monitors natural gas pumping stations.
COMMERCIAL	
BANKING:	Reads, analyzes and tabulates check data and monetary transactions in real-time applications at branch offices.
ACCOUNTING:	Acquires, processes and prints out man-hours-on-the-job for job/functions/time evaluation.
ENVIRONMENT CONTROL:	
CHEMICAL	Controls and monitors environmental condition throughout large buildings.
PLASTICS:	Stress analysis of plastic materials.
EXPLOSIVES:	Data acquisition and analysis of explosive shock waves.
DYES:	Monitoring and control of dye processing.



6

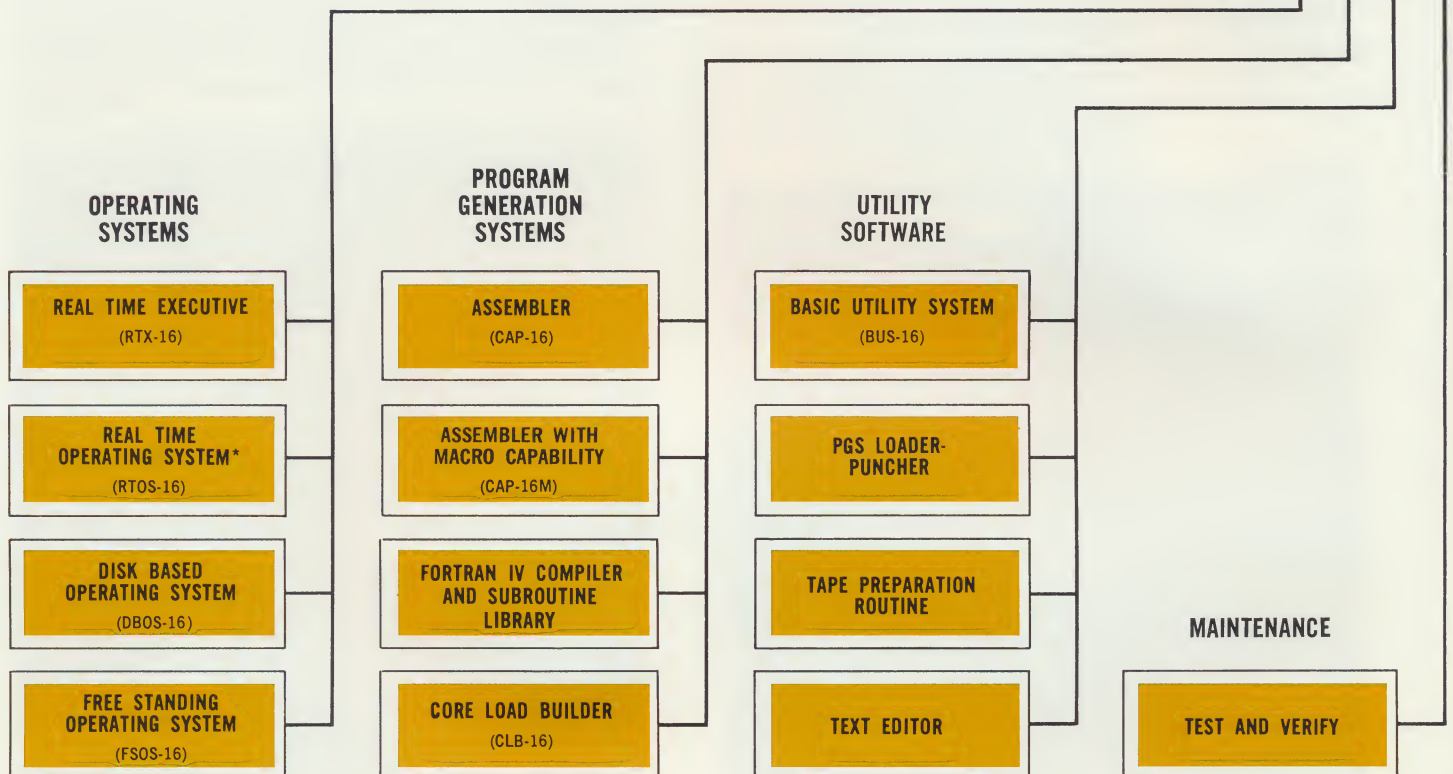
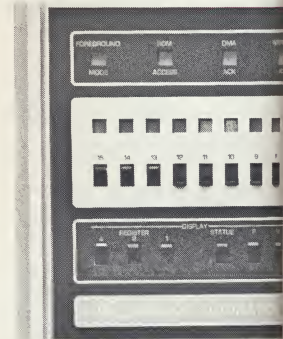
SOFTWARE

The SPC-16 system includes an extensive "Real-Time Systems Software Package" that greatly simplifies the user's task of program generation, system integration and facilitates project completion. These software packages provide programming ease and operating efficiency not found in any computer in the comparable price range.

The GA software library offers a wide selection of optional service-oriented software and functional-application "Automate" programs.

GA service software permits the user to write, edit, debug, and implement his programs quickly and economically. Service software includes Real-Time Executive systems, Disk based Operating System, Real-Time Operating System and Full FORTRAN, Logical and Physical Peripheral Drive programs, and specialized Programs Logic Modules.

The GA "Automates" are compatible with and run under the control of the Real-Time Executive service routines. They comprise general application programs designed to serve a broad range of processes (analog, digital, and communications I/O), and need only slight adaptation to suit a particular process within that range. With this approach, GA programmers can create specialized software for a particular project with a minimum expenditure of time, money, and manpower. Typical GA "Automates" include Local Process Monitoring, Remote Process Monitoring, XYZ Controller, Digital Step Controller, and Binary Synchronous Communication programs.



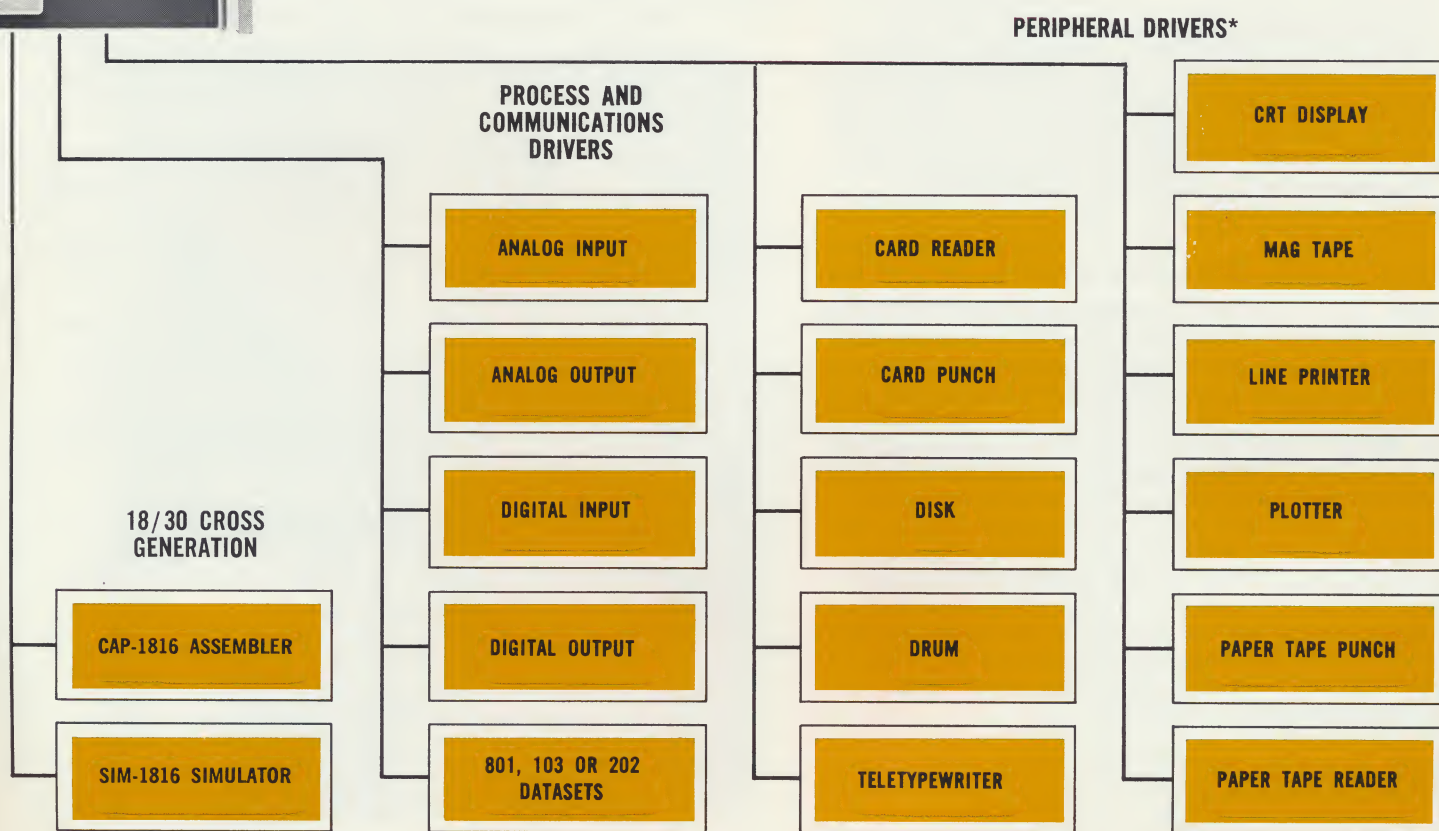
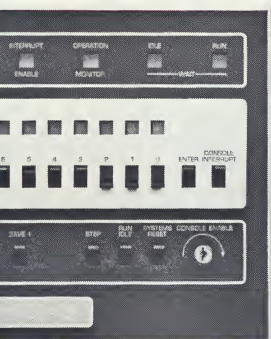
PROGRAM PHILOSOPHY

The SPC-16 instruction set and memory addressing capabilities assure that programs within the system will provide:

- Re-Entrancy: References to applicable program variables using base-relative addressing.
- Pure Procedure: All applicable program instructions and constants can be non-modifiable.
- Self-Relocation: Program references will be relative to the program counter whenever possible.

DEVICE INDEPENDENT I/O (LOGICAL AND PHYSICAL DRIVERS)

Device independence allows the user to specify input/output operations, regardless of the external device, by assigning a logical device that designates the physical device. Assignment of physical devices is performed at program execution and/or load time, rather than at program compilation/assembly time.



* Available only with associated hardware.

8

CROSS OPERATION SYSTEM

Programs for the SPC-16 can be generated on the powerful System 18/30. Time and effort of program generation are minimized if the System 18/30 has a complete complement of peripherals. A single-program generation facility further lowers your automation costs by reducing the amount of maintenance and spares, and the number of peripherals required on each computer.

The GA Cross Operating System consists of a cross assembler and a simulator program. The 18/30 Cross Assembler processes programs written in SPC-16 Macro Assembly language and generates a binary object program which may be loaded into the SPC-16 or System 18/30 via the SPC-16 Simulator program. The 18/30 Cross Assembler is fully compatible with the SPC-16 Program Generation System.

The SPC-16 Simulator simulates SPC-16 execution of the binary object programs on the 18/30. The Simulator contains a debug program that is identical in function to the SPC-16 Basic Utility System.

REAL TIME EXECUTIVE (RTX -16)

RTX-16 is a multi programming executive that provides supervisory control for real time programming on the SPC-16 computer. RTX-16 provides scheduling, memory allocation, input/output control and other functions required for real-time program operations. Programs can be written for use with RTX-16 using either the SPC-16 Assembly Language or Real Time FORTRAN. Source programs can be assembled and compiled off-line and built into RTX programs by the Core Load Builder. Programs are loaded on the disk by the RTX Disk Loader Utility. The modular construction of RTX permits the user to select only those features which are required for his particular system. New programs may be easily added for automation expansion requirements.

REAL TIME OPERATING SYSTEM (RTOS-16)

RTOS-16 is a multiprogramming foreground/background system for real time operation. The real time foreground programming provides scheduling, memory allocation, input/output control and other functions required for real-time program operations. Programs can be written for the RTOS-16 using either the SPC-16 Assembly Language or Real-Time FORTRAN. RTOS-16 contains background capability that allows source programs to be assembled or compiled and built into real time programs using the Core Load Builder. RTOS-16 also contains background processing that provides device independent input/output, sequential job processing, disk file management and program generations and executions. An interpreter is provided to assist debugging of foreground programs. This program prevents programming errors from disturbing foreground operations.

DISK BASED OPERATING SYSTEM (DBOS-16)

DBOS-16 is a device-independent disk operating system that provides the user with the capability of executing stacked jobs. The job stack can be a mixture of compilations, assemblies, inter-device data transfer, editing, etc. Devices are accessed through device-independent logical I/O system. An assembler and FORTRAN compiler are included to run under DBOS-16.

FREE STANDING OPERATING SYSTEM (FSOS-16)

GA's Free Standing Operating System (FSOS) is designed to simplify operations for non-disk systems. The system is built on-site from GA-supplied components for generating, loading, and debugging software. Two levels of software are available: (1) TTY-Only FSOS and (2) Device-Independent FSOS.

ASSEMBLER (CAP-16)

This symbolic assembler permits the user with a basic system configuration (TTY) and 4K of memory to generate programs with a minimum of effort. It allows the programmer to code programs using mnemonic operation codes and symbols to define machine instructions. A set of directives facilitate memory allocation, register utilization, definition of constant and literal data, and conditional assembly.

Programs can be either relocatable or absolute. Actual instruction addressing (base relative, program relative, or absolute) are automatically generated through usage of data and program section directives.

MACRO CAPABILITY (CAP-16M)

Expanding the system to include 8K of memory and high speed input/output devices substantially increases the systems performance. It provides the user with a powerful software tool for generating operation programs. It allows the programmer to code programs using mnemonic operation codes and symbols to define machine instructions. A set of directives facilitate memory allocation, register utilization, definition of constant and literal data, and conditional assembly. It also provides for programmer-defined MACROS, System input-output MACROS, external symbol reference and definition. Directives allow program linking by the Core Load Builder Program.

Programs may be either relocatable or absolute. Actual instruction addressing (base relative, program relative, or absolute) may be generated automatically through usage of data and program section directives.

FULL FORTRAN — REAL TIME FEATURES

The GA Real-Time FORTRAN system enables the programmer to solve problems using a higher level language.

This Real Time FORTRAN System produces object programs from source statement written in FORTRAN language. The SPC-16 FORTRAN meets the American Standard Full FORTRAN X3.9 -1966 with a wide range of extensions including:

In-Line Assembly Language	Logical Operations
Bit and Byte Arrays	Relational Expressions

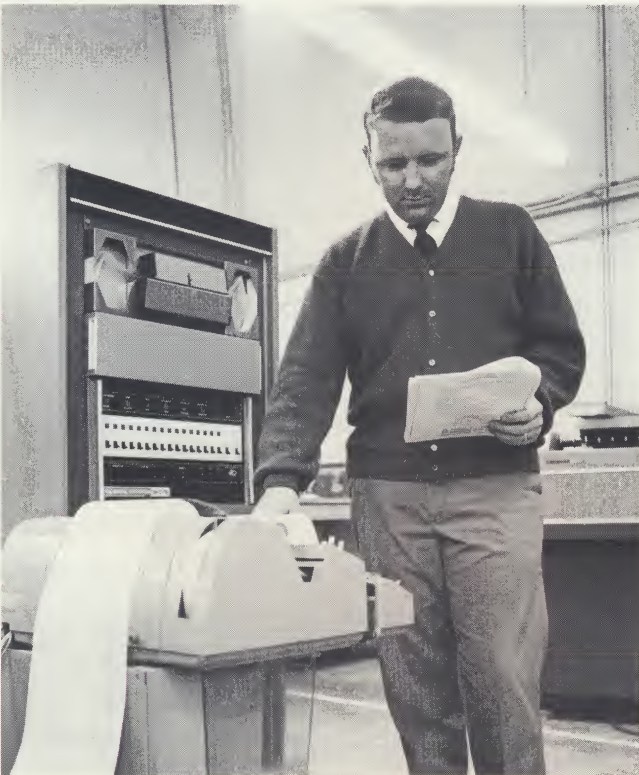
SUBROUTINE LIBRARY

The subroutine library consists of the USASI Basic External and Intrinsic Functions, together with routines for performing floating point and extended precision integer arithmetic operations.

CORE LOAD BUILDER (CLB-16) (LINKING LOADER)

The Core Load Builder processes user object programs. It performs all program linkages, selectively loads programs from the library, and produces an absolute object program output or loads directly to core for execution. The program relocation base, data area, and common area may be specified at load time with control statements.

A map and object tape may be produced as directed by the operator.



BASIC UTILITY SYSTEM (BUS-16)

BUS consists of a number of subroutines designed to aid the programmer in testing and debugging programs as well as performing the utility functions. Features included are:

Select Octal or HEX Mode of Operation	Load a Program Tape
Display & Change Memory	Execute a Program
Display & Change Registers	Breakpoint
Punch Object Tape	List Program

PGS LOADER-PUNCHER

The PGS Loader-Puncher allows the user to output selected areas of memory in object program format and to load object programs into memory. It will load object programs produced by itself or those produced by FORTRAN, CAP-16, BUS, and the Core Load Builder. The program relocation base and data area may be assigned by the user.

TAPE PREPARATION ROUTINE (TAPREP)

The TAPREP is a stand alone program designed to assist the user in the preparation and editing of symbolic source language input to the SPC-16 Assembler program. The user has the ability to delete, add, or modify a source statement when using TAPREP. TAPREP stores all source statements in memory.

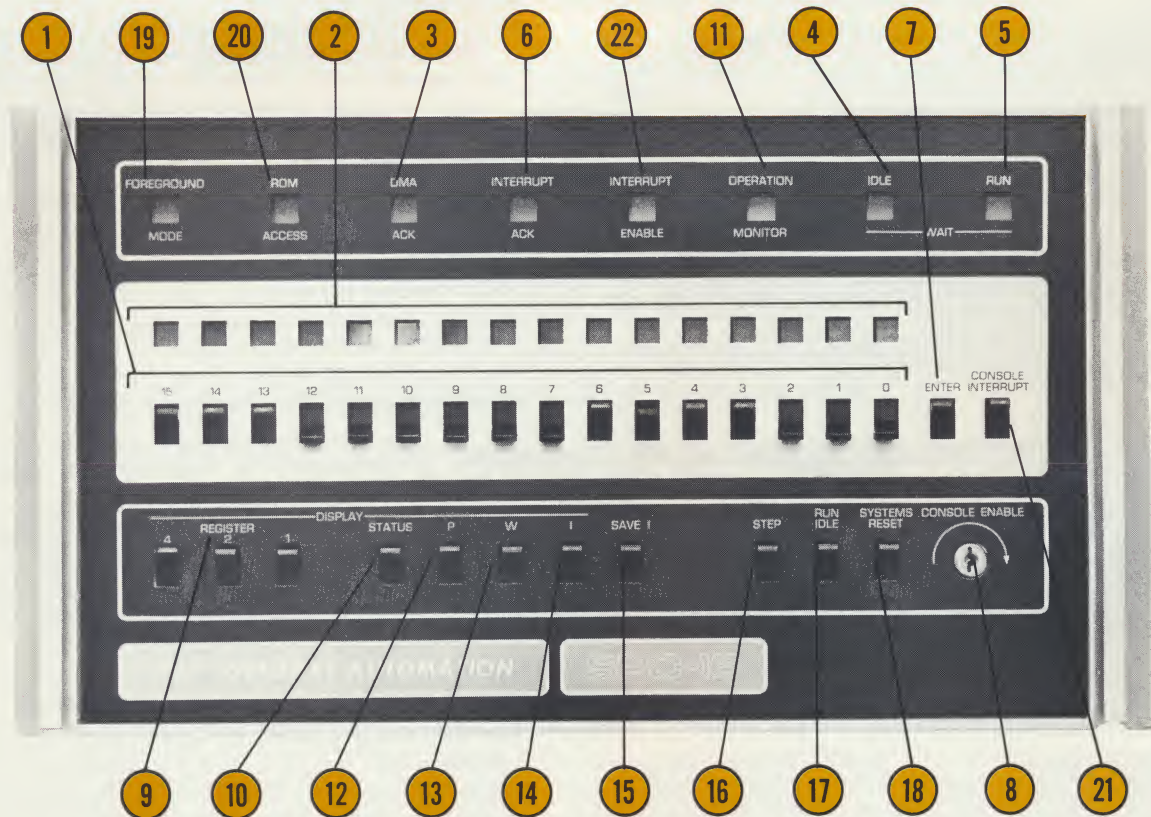
TEXT EDITOR

The Text Editor provides the paper system user with a convenient and flexible source text generation system. Source statements are entered via the keyboard or the tape reader. The entered source text may be output, statements added, deleted or modified. The entire text is maintained in core. The Text Editor permits the order of statements or groups of statements to be altered at any time. The final text is output to paper tape for use as input to the assembler or compiler.

TEST & VERIFY (T&V)

T&V programs enable the user to determine whether the hardware is functioning properly or to isolate a malfunction if one should occur. T&V programs are provided for Processor, High Speed Paper Tape Reader, High Speed Paper Tape Punch, Card Reader, Card Punch, Line Printer, Teletypewriter, Mag Tape, Disk, and SPC-16 System Test.

SYSTEM CONSOLE



1. Sixteen Console Data switches used to enter data or instructions into the console display register and the selected register. These switches may be tested as sense switches or used as data input switches during program execution.
2. Register Display indicators indicates contents of the console display register. The register is loaded under program control in the run mode and by the selected register in the idle mode. Least significant 4 bits converted to one of 16 voltage levels for use with the Autovue feature.
3. Direct Memory Access/Automatic Data Channel Acknowledge indicator. Indicates that the DMA is stealing a cycle.
4. Idle Indicator. Indicates when the computer is in the idle mode. When both Run and Idle indicators are lit, the computer is in the wait mode.
5. Run Indicator. Indicates when the computer is in the run mode. When both Run and Idle indicators are lit, the computer is in the wait mode.
6. Interrupt Acknowledge indicator indicates when an interrupt is being serviced.
7. Enter switch, in the idle mode, replaces the contents of the selected register with the contents of the console data switches.
8. Console Key Lock used to disable all console switches when in the locked position. Console can also be locked in the idle mode.
9. Three Register Address switches used to select one of the eight operational registers to receive data from the console

- data switches. The selected registers value will be displayed on the console indicator lights in the idle mode.
10. Status switch used to place the computer status indicators and shift counter into the System Console Display register.
11. Operations Monitor Alarm indicates when the alarm has been activated.
12. "P" switch displays the contents of the program counter on the console display indicators. The program counter may be changed by the console data switches.
13. "W" switch displays the contents of the Working register.
14. "I" switch displays the contents of the instruction register.
15. Save "I" switch used to preserve the contents of the instruction register for repeated execution in the idle mode.
16. Step switch used to execute the instruction currently in the "I" register and increment the program counter to the next instruction. This switch starts program execution after the computer is placed in the run mode.
17. Run/Idle switch used to place the computer in either the idle or run mode.
18. System Reset switch used to initialize the computer and the input/output system.
19. Foreground/Processing indicator indicates which set of 8 programmable registers is being used.
20. ROM Access indicator indicates when a Read Only Memory is being accessed for instructions or data.
21. Console Interrupt switch requests an interrupt.
22. Interrupt Enable indicator indicates interrupt system is enabled.

COMPUTER POWER

Both standard and special hardware features are uniquely combined in the SPC-16 to provide unprecedented "computer power" capability.

Efficient, Powerful Processor Power:

- real-time operation and control
- macro ROM bus and control
- micro ROM bus and control
- external processing control
- direct computer control and bus
- powerful bit/byte/word-complete instruction set
- direct memory access bus
- programmable I/O bus
- systems console I/O bus

Versatile Addressing Power:

- direct access to any word in memory with only one instruction
- addressing of any byte in memory or any bit in memory with just one instruction
- program relative and base relative addressing
- indirect or indexed addressing

Fast Response Power:

- minimum reaction time to automatic priority interrupts
- foreground/background processing
- load and store of all registers and status data with one instruction

Flexible, High Speed Communications Power:

- programmable parallel I/O channel
- systems console parallel I/O channel
- direct memory access port (standard)
- multiplexed data channels at 250 KHZ transfer rate
- 1.04 MHZ data channel
- arithmetic and logic function capabilities

ROM Power for Safe, Protected Programming:

- completely interchangeable ROM and R/W memory modules
- instructions operable in both R/W memory and ROM
- 16 general-purpose registers
- string processing on registers from R/W memory or ROM
- overlapped processing with ROM
- base-relative and program-relative instructions
- versatile execute instructions

ROM WITH SPC-16

For ROM to be of true value to the user, it must improve instruction execution time as well as protect programs. It must also permit the user to build his own ROM and to do pure procedural programming. To achieve this, the computer must provide a variety of features: an execute instruction, program and base relative addressing, multiple general-purpose registers, fast ROM instruction execution time, physically interchangeable ROM and R/W memory modules, instructions that can be executed from either ROM or R/W memory, and relocatable auto-restart interrupt vectors.

GA's SPC-16 has been carefully organized to incorporate all of these features, permitting full and effective use of ROM. Special design points which optimize the use of ROM with the SPC-16 include:

Versatile Execute Instruction allows the content of any register to be executed as an instruction, thereby eliminating the need for program modifications. This allows modification and execution by a program in ROM.

Base and Program Relative Addressing allows ROM programs to use all of memory. Programs operate relative to the program counter in ROM; data tables can be addressed relative to the Base Register. Data tables can be anywhere in core memory and data can be relocated in core by changing the value placed in the Base Register. This internal time-sharing feature is especially valuable in controlling a number of process steps with a single machine. For this type of operation, basic program instructions suitable for all configurations are stored in ROM. Specific process instructions are entered into R/W memory to be handled as data.

Sixteen General-Purpose Registers provide high speed storage capability for programs executed from ROM. They also speed information transfers in and out of R/W memory and out of ROM. Also, the amount of time-consuming memory access needed during program execution is greatly reduced. The SPC-16 is capable of arithmetic and logic of performing register to register operations in a single command without reference to memory.

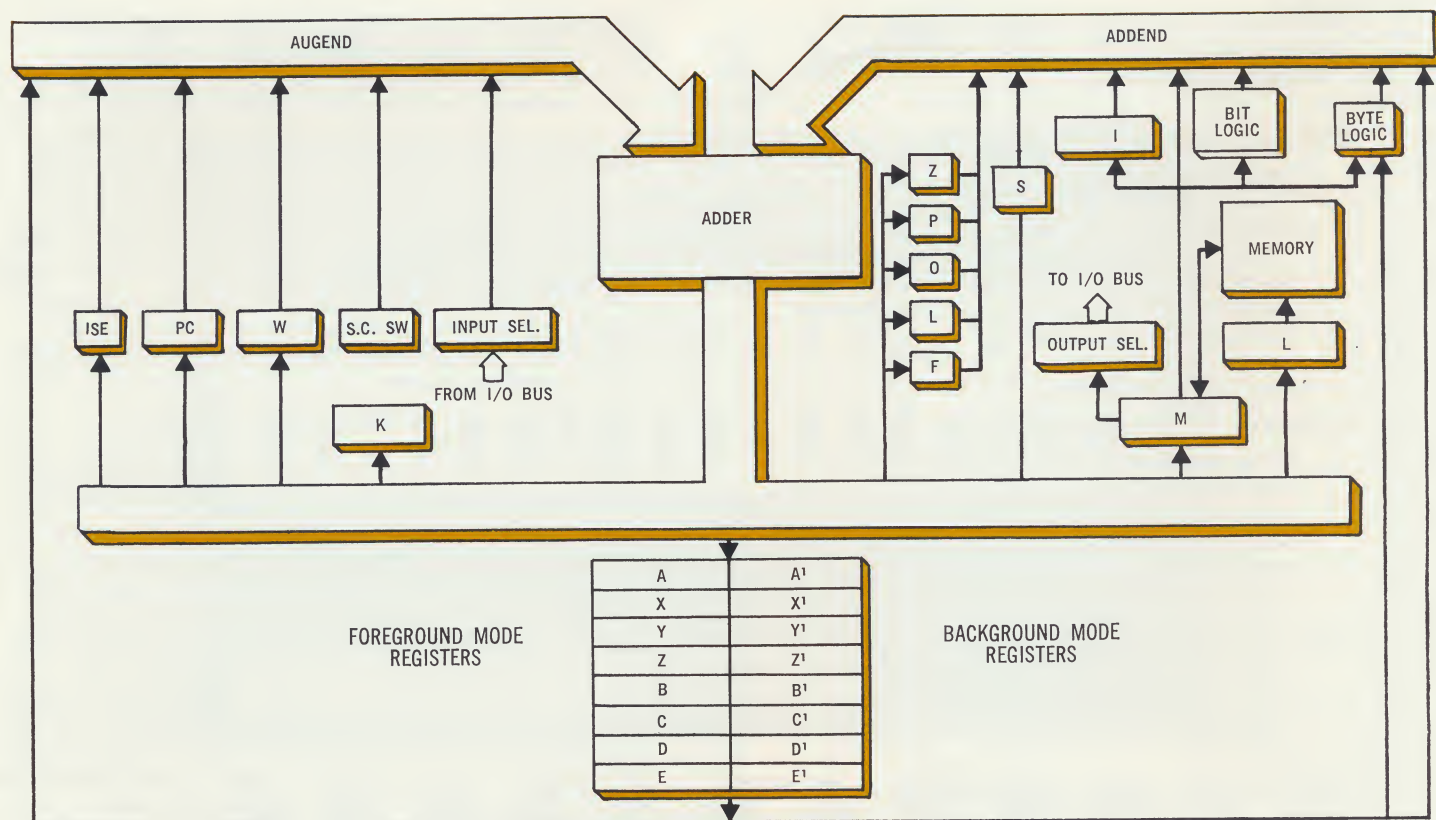
Fast Execution Time — ROM is 480 nanoseconds fast; it speeds program execution time and enhances real-time computation and control.

Entire Instruction Set Operates in Both ROM and R/W Memory, thus reducing program development and debugging time. Requirements for specialized programs are minimized for each hardware configuration and facilitates transfer of programs from core to ROM. Any location in memory can be addressed. Execution times change automatically according to the memory being addressed.

Completely Interchangeable Memory Modules provide expansion and relocation flexibility, allowing memory configurations to fit specific applications.

Relocatable Power-Fail/Auto-Restart Interrupt Vectors allow ROM to be used to protect a loader program, enabling the computer to automatically reload the main program after a power failure. If ROM is located in one of the eight memory positions, the interrupt vector can be reassigned to that position.

ORGANIZATION



DATA FLOW ORGANIZATION

ADDRESSING

The SPC-16's powerful addressing system combines eleven addressing techniques with eleven instruction classes. Memory may thus be addressed as zero to 32K Words, zero to 65K Bytes, or zero to 512K Bits or flip-flops. Word, byte, and bit addressing instructions may be freely interleaved in the program, since each instruction has its own addressing mode and range.

Program-relative and base-relative addressing in the SPC-16 enables pure procedure programming as well as effective use of ROM programming. With the exception of program-relative addressing, all addressing modes of the SPC-16 can be either single or double word. Addressing flexibility also permits the re-entry of subroutines, allowing for subroutine nesting.

The eleven addressing modes comprise: Direct; Direct, Indexed; Indirect; Indirect, Indexed; Program Relative; Program Relative, Indirect; Base Relative; Base Relative, Indexed; Base Relative, Indirect; Base Relative, Indirect, Indexed; and Literal.

DIRECT:

The Effective Operand Address is the displacement of the instruction. The address may be modified by a selected index register; the modified address then becomes the Effective Operand Address. If the Direct Address instruction is double word, the Effective Operand Address is the content of the next memory location ($P+1$), and the program counter is automatically incremented to ($P+2$). Then this new address may be modified by a selected index register.

INDIRECT:

The Effective Operand Address is developed from the contents of the indirectly-addressed word. The new operand address may be the Effective Operand Address, or may be modified by a selected index register; the modified address then becomes the Effective Operand Address. If the Indirect Address instruction is double word, the next word ($P+1$) is the first of a two-level indirect instruction, and the program counter is automatically incremented to ($P+2$).

UNIT	DESCRIPTION	FUNCTION
ISE	Interrupt System Enable	Controls the program interrupt system. Is controlled by interrupt enable, interrupt inhibit, restore interrupt system enable, and jump-to subroutine instructions.
P	Program Counter Register (15 Bit)	Holds address of the instruction being executed. Automatically incremented during instruction execution; also, can be changed by jump, skip, and jump-to subroutine.
W.	Working Data Register (16 Bit)	For most memory referenced and memory referenced indexed instruction, it maintains the effective operand address.
INPUT/OUTPUT BUS	Input/Output Bus (16 Lines)	Contains data being input from or output to external devices.
SYSTEM CONSOLE SWITCHES	System Console Switches (16)	Permit the programmer to manually input addresses, instructions, or data from the system console.
K	System Console Display Register (16 Bit)	Displays the contents of the selected register (16 programmable registers, status indicators, P, W, or I). May also be loaded under program control using the DSPL instruction. Voltage representation of its contents is outputted to the real-time visual debug line.
Z	Zero Status Indicator (1 Bit)	Indicates the zero or non-zero results of the last arithmetic operation.
P	Positive Status Indicator (1 Bit)	Indicates the plus or minus results of the last arithmetic operation.
O	Overflow Status Indicator (1 Bit)	Indicates an arithmetic overflow during all arithmetic instructions.
L	Link Status Indicator (1 Bit)	Indicates a carry out of Bit 15 during arithmetic operations, and is a single Bit register for shifts.
FORE-GROUND MODE	Foreground Mode Indicator	Controls the selection of the two sets of general-purpose programmable registers.
I	Instruction Register (16 Bit)	Contains the instruction being executed.
M	Memory Data Buffer Register (16 Bit)	Contains data or instruction being read from or written into memory; buffers all output transfers.

UNIT	DESCRIPTION	FUNCTION
BIT LOGIC	Bit Logic	Provides bit handling capability of SET, BIT, RESET BIT, TEST BIT instructions.
L	Memory Location Register (15 Bit)	Contains the address of data or instruction being accessed from memory.
S	Shift Counter Register (4 Bits)	Controls the number of shifts performed for each of the shift instructions.
BYTE LOGIC	Byte Logic	Provides the byte handling capability of LOAD BYTE, STORE BYTE, EXCHANGE BYTE, ZERO BYTE, etc., instructions.
A/A'	General-Purpose Register (16 Bit, programmable)	Can be an accumulator, source, or destination register for arithmetic and logic functions. Holds address of data being output or inputted. Holds data being inputted or outputted. A or A' selection is dependent upon the state of the Foreground Mode Control.
X/X'	Index/General-Purpose Register (16 Bit, programmable)	Index register. Can also perform functions identical to "A/A'".
Y/Y'	Index/General-Purpose Register (16 Bit, programmable)	Index register. Can also perform functions identical to "A/A'".
Z/Z'	Index/General-Purpose Register (16 Bit, programmable)	Index register. Can also perform functions identical to "A/A'".
B/B'	General-Purpose Register (16 Bit, programmable)	Functions identical to "A/A'".
C/C'	General-Purpose Register (16 Bit, programmable)	Functions identical to "A/A'".
D/D'	Base-Relative Addressing and General-Purpose Register	Generally contains the base address when using base-relative addressing; can be used as a general-purpose programmable register identical to "A/A'".
E/E'	Subroutine Linkage and General-Purpose Register	During the execution of a subroutine, contains the return address and Interrupt Status Enable Control and Indicator bit. Can also be used identical to "A/A'".
ADDER	16 Bit Binary, Full Adder	Performs all arithmetic functions. Inputs data from the Augend & Addend, and outputs to the Data bus.

BASE RELATIVE:

The Effective Operand Address is the sum of the content of the base register and the displacement field. This new operand address may be modified by an indirect or indexed modifier, or both. When using both, the indirect modification occurs prior to adding the selected index register. If Base-Relative addressing is double word, the Effective Operand Address is the contents of the next memory location (P+1) added to the base register, and the program counter is automatically incremented to (P+2).

INDEXING:

Three hardware Index Registers are standard features of the SPC-16; three more can be added as part of the Foreground/Background Processing option. Any of these registers may be selected for those instructions whose address fields can be indexed. When indexing is specified, the content of the selected register is added to the address field of the instruction, thus forming the Effective Operand

Address. Indexing in the SPC-16 is post-indexing, occurring only after indirect and base-relative modifications have taken place.

PROGRAM RELATIVE:

Program-Relative instructions are always single word. The Effective Operand Address is the content of the program counter and the displacement field of the instruction. When an indirect modification is specified, this new Effective Operand Address becomes the content of the location (P+1 + Disp).

LITERAL:

The SPC-16 has two classes of instructions for specifying Literal Addressing: Register Operate Literal and Register Operate Compare Literal. The second operand of the instruction is the content of the memory location immediately following the location of the current instruction (P+1), and the program counter is automatically incremented to (P+2).

INSTRUCTION CLASSES/REPertoire

The SPC-16's extensive and powerful instruction set efficiently handles bits, bytes, and words in both ROM and R/W memory, thus reducing programming and execution time as well as memory requirements for real-time applications. Subroutines and interrupt subroutines are executed via instruction which load and store all registers and indicators. Jump-to and exit-from subroutines require only one memory cycle each. Interrupt enable and inhibit control are also provided, with automatic inhibit upon an interrupt acknowledge or other subroutine entry. Upon subroutine exit, the enable or inhibit status is restored.

In addition, for list/table processing, the SPC-16 provides eleven modes of operand addressing to the load, store, compare, load byte, store byte, increment memory, replace instructions, set bit, reset bit and test bit.

MEMORY REFERENCE

There are 16 instructions in this class, containing all the instructions that reference memory except I/O. These instructions can store any or all programmable registers in memory, load any or all programmable registers from memory, compare any register with memory, increment the contents of any memory location, or cause a jump to any location within memory. The memory can be referenced, as 0-32K words, 0-65K Bytes, or 0-512K Bits or flip-flops, depending on the instruction used.

This instruction class has the flexibility to use single or double word instructions, giving it the ability to access all of memory with one instruction. The complete range of addressing (direct, direct-indexed, indirect, and indirect-indexed) is used by this class of instruction.

The execution of a subordinate link instruction provides the program counter with a new memory reference location and automatically stores the previous program counter value (P+1) in the "E" register. The system interrupt enable status is also stored in Bit 15 of the "E" register.

OP CODE	V.P.	DESCRIPTION	CYCLE
MEMORY REFERENCE (4)			
LDA	Addr	Load Register A	2
STA	Addr	Store Register A	2
JMP	Addr	Jump Unconditionally	1
JSR	Addr	Jump to Subroutine	1

MEMORY REFERENCE WITH INDEXING (12)			
LDR	R, Addr, X	Load Register	2
STR	R, Addr, X	Store Register	2
CMR	R, Addr, X	Compare Memory with Register	2
LDBY	R, Addr, X	Load Byte	2
STBY	R, Addr, X	Store Byte	2
LARS	Addr, X	Load All Register and Status	10
SARS	Addr, X	Store All Register and Status	10
INCM	Addr, X	Increment Memory	2
DECM	Addr, X	Decrement Memory	2
SBIT	n, Addr, X	Set Bit n (n = 0 - 7)	2
RBIT	n, Addr, X	Reset Bit n (n = 0 - 7)	2
TBIT	n, Addr, X	Test Bit n (n = 0 - 7)	2

SKIP

There are eight instructions in this class; each instruction tests one of 4 indicators for either a true or false condition. If the condition is met the "Disp" is used to form an effective address (P±256) for the location of the next instruction. If the condition is not met the next instruction (P+1) is executed.

OP CODE	V.P.	DESCRIPTION	CYCLE
SKIP (EXTENDED MNEMONICS) (8)			
SKZ	DISP	Skip if Zero	1
SKN	DISP	Skip if Non Zero	1
SKP	DISP	Skip if Plus	1
SKM	DISP	Skip if Minus	1
SKOT	DISP	Skip on Overflow True	1
SKOF	DISP	Skip on Overflow False (No Overflow)	1
SKS	DISP	Skip on Link Set	1
SKR	DISP	Skip on Link Reset	1

REGISTER CHANGE

There are 16 instructions in the register change class. Each instruction specifies one of the programmable register and the operation to be performed on the contents of that register. The operation is performed on the contents of the selected register and the results replace the original content of the register. The resultant zero, plus, overflow and link conditions are placed into their indicators.

OP CODE	V.P.	DESCRIPTION	CYCLE
REGISTER CHANGE (16)			
ZERO	R	Zero Register	1
XEC	R	Execute Register Contents	1
TRS	R	Transfer Register to Status	1
TSR	R	Transfer Status to Register	1
RCSW	R	Read Console Switches	1
DSPL	R	Transfer Register to Console Display	1
ZLBY	R	Zero Left Byte	1
ZRBY	R	Zero Right Byte	1
EXBY	R	Exchange Bytes	1
RTRN	R	Subroutine Return	1
RISE	R	Restore Interrupt System Enable	1
RLK	R	Add Link to Register	1
INCR	R	Increment Register	1
DECR	R	Decrement Register	1
CMPL	R	Complement Register	1
ADD	R	Shift Counter to Register	1

SHIFT

There are four instructions in this class. Each instruction can select any of the programmable registers, the shift count number, and the type of shift. In each case the shift proceeds bit by bit until the shift count has been satisfied with the exception of the Shift Right Logical and Count which will terminate if a one is shifted into the Link indicator.

OP CODE	V.P.	DESCRIPTION	CYCLE
SHIFTS (8)			
SRA	R, Count	Shift Right Arithmetic	$(1 + n/2)$
SRC	R, Count	Shift Right Circular	$(1 + n/2)$
SRLC	R, Count	Shift Right Logical and Count	$(1 + n/2)$
SRCL	R, Count	Shift Right Circular Thru Link	$(1 + n/2)$

REGISTER OPERATE

There are 21 instructions in this class. Each instruction specifies a source register (Rs), destination register (Rd) and one of 11 Arithmetic, Logic or Transfer operations. If "OPERATE" is selected the operation is performed on the contents of (Rs) and (Rd) and the results are placed in (Rd). The Zero, Plus, Overflow, and Link conditions of the results are placed into the indicators.

If "OPERATE" and "COMPARE" is selected, the operation is performed and the conditions of the result are placed into the indicators. In this case, the (Rs) and (Rd) registers remain unaltered.

If literal addressing is selected the Source Register (Rs) becomes the next memory location (P+1). The results of the operation is placed in the Destination Register (Rd), otherwise the instruction execution is identical.

OP CODE	V.P.	DESCRIPTION	CYCLE
REGISTER OPERATE (6)			
RTR	Rd, Rs	Transfer Register	1
ADD	Rd, Rs	Add Registers	1
SUB	Rd, Rs	Sub Registers	1
AND	Rd, Rs	And Registers	1
XOR	Rd, Rs	Exclusive or Registers	1
OR	Rd, Rs	Or Registers	1

REGISTER OPERATE COMPARE (5)			
ADDC	Rd, Rs	Add Registers and Compare	1
SUBC	Rd, Rs	Subtract Registers and Compare	1
ANDC	Rd, Rs	And Registers and Compare	1
XORC	Rd, Rs	Exclusive or Registers and Compare	1
ORC	Rd, Rs	Or Registers and Compare	1

REGISTER OPERATE LITERAL (6)			
LDV	R, (Value)	Load Value into Register	2
ADDV	R, (Value)	Add Value to Register	2
SUBV	R, (Value)	Subtract Value from Register	2
ANDV	R, (Value)	AND Value with Register	2
XORV	R, (Value)	Exclusive OR Value with Register	2
ORV	R, (Value)	OR Value with Register	2

REGISTER OPERATE LITERAL COMPARE (4)			
ADDVC	R, (Value)	Add Value to Register and Compare	2
SUBVC	R, (Value)	Subtract Value from Register and Compare	2
ANDVC	R, (Value)	AND Value with Register and Compare	2
XORVC	R, (Value)	Exclusive OR with Register and Compare	2

EXTENDED PROCESSOR OPTION

There are two instructions in this class. The multiply instruction causes the 15 bits of the A Register to be multiplied by the specified number of bits in the B Register. The resultant 32 bits are placed in the B and C Registers; most significant bits in B, least significant bits in C.

The divide instruction causes the A Register to be divided into the B and C Registers. The quotient is placed in the C Register and the remainder in the B Register. The number of bits desired in the quotient is specified in the divide instruction.

OP CODE	V.P.	DESCRIPTION	CYCLE
EXTENDED PROCESSOR OPTION (2)			
MPY	A*B(n)→B,C	Multiplies Registers	$1.92 + 0.48 (n) \mu S$
DIV	(B,C)/A→C(n),B	Divides Registers	$2.40 + 0.96 (n) \mu S$

CONTROL

There are 9 instructions in the Control class. The "wait" instruction causes the processor to stop executing instructions but interrupts, and Data Channel requests can still be serviced. If the computer is equipped with Foreground/Background option (8 additional programmable registers) the Foreground/Background selection is made with the control instructions FMS or BMS.

OP CODE	V.P.	DESCRIPTION	CYCLE
CONTROL (9)			
INE		Interrupt ENABLE	1
INH		Interrupt Inhibit	1
FMS		Foreground Mode Set	1
BMS		Background Mode Set	1
PMA		Pulse Monitor Alarm	1
LKS		Set Link	1
LKR		Reset Link	1
SYNC		General Sync Pulse	1
WAIT		Stops Instruction Execution	1

PROGRAMMED INPUT/OUTPUT

There are six instructions in this class. These instructions control all programmed communication with external devices or processes. This instruction class has 4 main modes of operation input data, output data, test external devices, or control external devices.

In the input/output mode, any of the programmable registers may be selected and used as the source/destination register for data transfers. If input/output is to be made directly into or out of memory, the selected register contains the address of the memory location to be used.

OP CODE	V.P.	DESCRIPTION	CYCLE
INPUT/OUTPUT (6)			
DTOR	R, Device	Data Transfer Out of Register	2
DTIR	R, Device	Data Transfer Into Register	2
DTOM	R, Device	Data Transfer Out of Memory	2
DTIM	R, Device	Data Transfer Into Memory	2
CTRL	FUN, Device	Output Function Control Pulse	2
TEST	FUN, Device	Test Device	2

16

SYSTEM I/O

The Programmable I/O Bus, Direct Memory Access (DMA) Data Channels, and interrupt structure of the SPC-16 simplify problems inherent in real-time/on-line programming. Such applications require multiple, high-throughput I/O channels, many hardware priority interrupts, and minimum overhead and response time. In addition to fulfilling these requirements, the SPC-16 saves memory and time by performing arithmetic and logical functions on incoming data from the DMA Channel and the current contents of the memory locations.

PROGRAMMABLE I/O BUS:

The System Console TTY is the minimum I/O for the SPC-16 System. The TTY works on the Programmable I/O Bus, operating on a TTY "not busy" (level 6) interrupt basis. The TTY Controller, an integral part of the SPC-16, provides the necessary serial-to-parallel and parallel-to-serial conversion.

The parallel Programmable I/O Bus transfers data to or from memory or registers at a maximum rate of 520K words per second. This I/O bus can service up to 64 devices with over 200 control functions, and can be interfaced to any of the GA process, communications, and peripheral interface units.

DIRECT MEMORY ACCESS PORT:

The Direct Memory Access Port, providing high-speed data transfer directly into memory without interrupting normal program processing, Direct Computer Control are standard features of the SPC-16. Two options increase the DMA Port's extreme usefulness: (1) A Multiplexed Data Channel, (2) Single Data Channel(s).

Multiplexed Data Channel (MDC) provides eight data channels, capable of either single channel operation at 340 KHZ, or operation in an interlaced mode at 340 KHZ total. The registers associated with each channel are dedicated memory locations in lower core, and require only store instructions for initialization. These channels give an economical I/O interface for high-speed data processing peripherals such as disks, magnetic tapes, drums, etc.; and other processes that require direct memory access.

Single Data Channel (DC) provides transfer rates of up to 1.04 MHZ. Any number of DC's can be incorporated for combined operation in an interlaced mode at 1.04 MHZ total. DC hardware registers and control logic are located external to the processor to optimize the higher transfer rates and to let the DMA Port remain general. The single data channels operate on a priority basis to accommodate different peripheral speeds.

Direct Computer Control is a standard feature of the SPC-16. The DMA Channel permits arithmetic and logical operations to be performed on the incoming data and the current content of the addressed memory location. Standard arithmetic and logic functions performed with this option comprise: Read; Write; Read and Clear Memory; Add and Write; Sub and Write; Inc and Write; OR

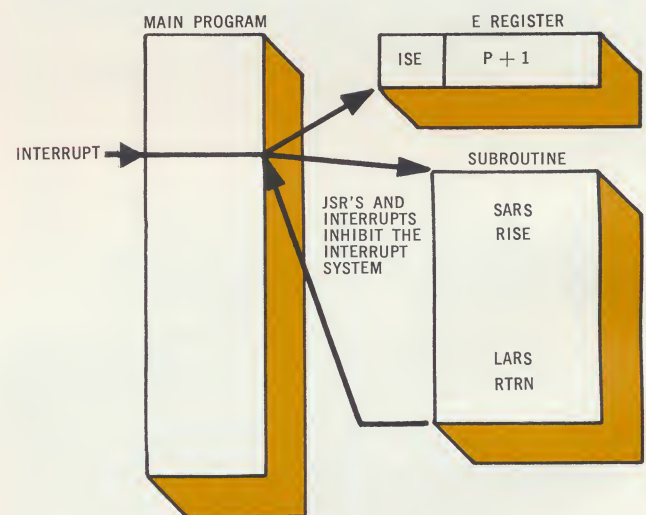
and Write; XOR and Write; and AND and Write. Special status indicators return the carry and zero results back to the data channel for control functions.

INTERRUPTS:

The SPC-16 priority interrupt system is designed for minimum response time and overhead in servicing interrupts. The basic system has five internal and three external priority interrupts. The external interrupts are expandable to an unlimited number of hardware priority levels. The programmer has hardware mask control over each interrupt level, in addition to a System Master Interrupt Enable and Inhibit instruction. For full system protection, the power-failure detection and auto-restart interrupt cannot be inhibited.

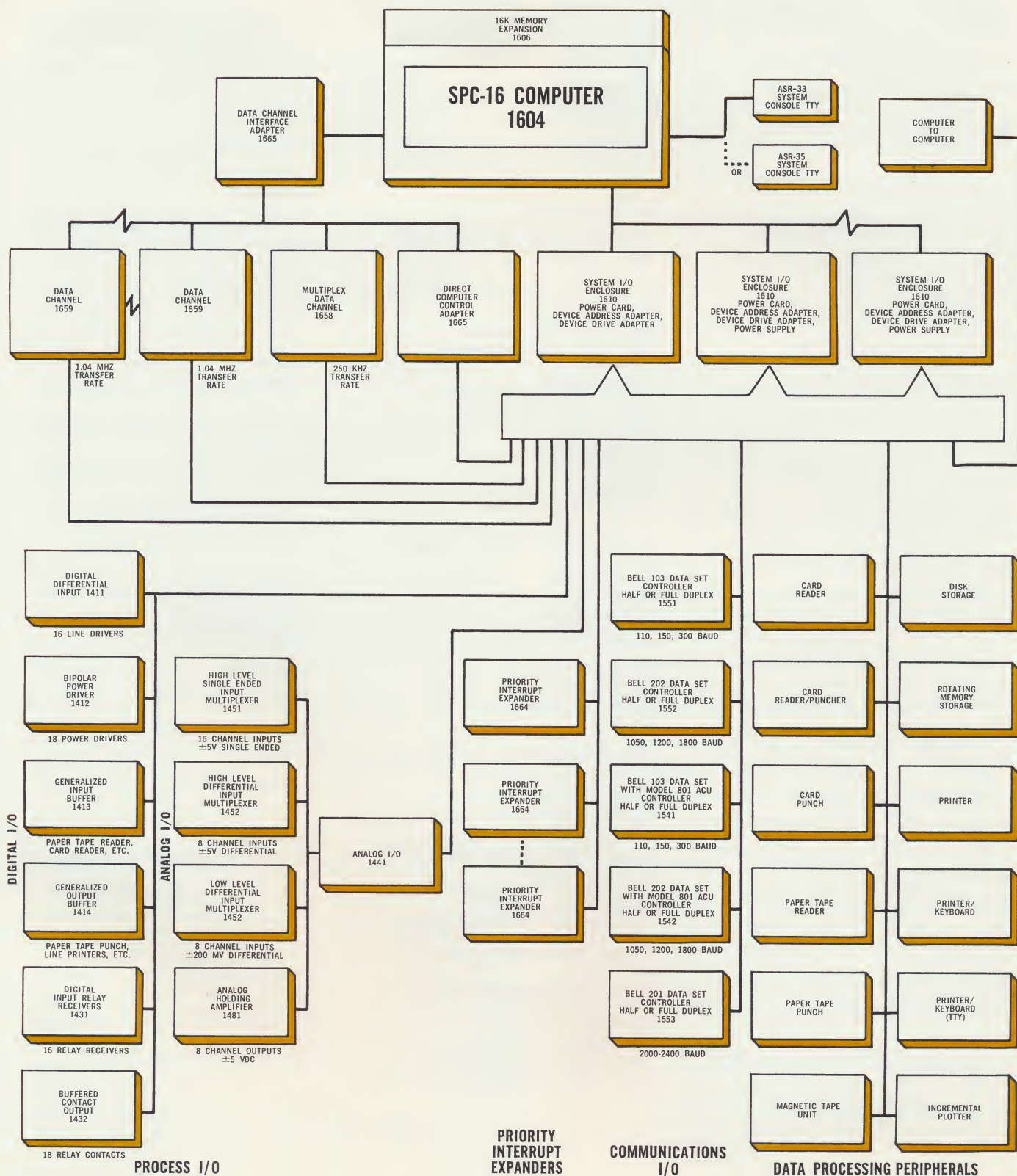
Interrupts are serviced via a forced "Jump-to-Subroutine" indirect (JSR) instruction through a dedicated memory location or vector. JSR automatically inhibits the interrupt system, saves the Return Address and Interrupt Enable/Disable status in subroutine linkage register (E). Return to the main program is achieved with a "Return" command which transfers the subroutine linkage register to the program counter and reinstates the interrupt status.

If higher priority interrupts are needed, the master interrupt system must be enabled and the proper masking performed. If more than one level of interrupt is required, the programmable registers may be saved through the single instruction "Store All Registers and Status" (SARS). After interrupt completion, the single instruction, "Load All Registers and Status" (LARS) completely restores all status and programmable registers. For only one level of interrupt, the "Foreground/Background Process" mode can service the interrupt routine. This mode provides minimum overhead and fast response time, yet gives the main program complete protection.



INTERRUPT AND SUBROUTINE
LINKING PROCEDURE I/O

SYSTEM SUMMARY



SPECIFICATIONS

TYPE:

General-purpose automation computer
Read/Write and Read Only memories
(interchangeable)
Parallel binary two's complement
arithmetic
Single and double-word addressing
Programmed and direct memory access
input/output

CONSTRUCTION:

High-speed TTL Integrated Circuits (I/C)
Large-scale Integrated Circuits (LSI)
Medium-scale Integrated Circuits (MSI)
Small-scale Integrated Circuits (SSI)
Large-board technology
Totally wire-free; no slot sensitivity of
processor board or memory stacks
Patented vertical bus-bar power
distribution system (Provides
multi-layer benefits, low noise)
Completely field expandable
Complete field interchangeability of all
parts

MEMORIES:

Read/Write (R/W Memory):
Random access
960-nanosecond cycle time
Field expandable to 32K (4K increments)
Wide-temperature lithium ferrite core
memory, 3D-3 wire
Independent current regulation for each
memory stack
Independent strobe timing for each
memory stack
Organization:
16-bit data word (32K max.)
8-bit data byte (65K max.)
1-bit data bit (512K max.)

Read Only Memory (ROM):

Random access
480-nanosecond cycle time
Organization:
16-bit data word (32K max.)
8-bit data byte (65K max.)
1-bit data bit (512K max.)
Interchangeable with R/W memory
512, 1024, 2048-word increments

REGISTERS:

Hardware registers	25
Programmable registers	18
Accumulators	16
Index/accumulators	6
Base register	2
Subroutine linkage	2
Shift counter register	1
Console display register	1

STATUS INDICATORS:

Zero
Plus
Link
Overflow
Interrupt enable
Foreground mode

ARITHMETIC AND LOGIC:

Parallel binary, two's complement, fixed
point
Bit, byte and word
ADD, SUB, COMP, INC, DEC, AND,
OR, XOR, NEG, SET BIT, RESET
BIT, TEST BIT

INSTRUCTIONS (Eleven Classes)

Memory reference
Memory reference indexed
Skip (8) ± 256 locations on eight
conditions
Register operate
Register operate and compare
Register operate literal
Register operate literal and compare
Register change
Shift (0-16 bits)
Control
Input/Output (addressing to 64 devices,
single-word instruction)

ADDRESSING (Eleven Modes):

Direct
Direct, indexed
Indirect
Indirect, indexed
Program relative
Program relative, indirect
Base relative
Base relative, indexed
Base relative, indirect
Base relative, indirect, indexed
Literal

INPUT/OUTPUT:

System Console TTY:
16-bit parallel bus
Parallel-to-serial internal controller (for
Console TTY)

Direct Memory Access Bus:

DMA bus standard
Expandable to multi-channel operation
with priority sequencing
1 MHZ transfer rate (cycle stealing)
(DC)
250 KHZ transfer rates (cycle stealing)
(MDC)

Standard DMA automatic data channel:

Transfers data into and out of memory
Arithmetic and Logical DMA (Option)
In addition to data transfers,
arithmetic and logic functions can be
performed between incoming data and
the contents of memory.

Programmed Parallel I/O Bus:

16-bit parallel I/O bus
64 external devices, over 200 control
functions
520 KHZ (max.) transfer rates
Data transfer:

To/from memory
To/from any one of the sixteen
programmable registers

Interrupts

Unlimited priority interrupt levels (PIE)
Programmable system interrupt control
Individual hardware mask on each
priority interrupt

8 internal interrupts:

Powerfail
Auto-Restart
Real-Time Clock
Console TTY Ready
Console TTY Not Busy
Console interrupt
External priority interrupt (2)

DIAGNOSTIC DEBUG DISPLAY ("AUTOVUE"):

Voltage representation of digital pattern
in the console display register and scope
synchronization pulse
(Allows visual display and
examination of the process, the
program, and the computer.)

DIMENSIONS:

Height: 10.5 inches
Width: 19.0 inches
Depth: 23.6 inches

WEIGHT:

80 pounds (maximum)

TEMPERATURE:

Operable at 0°C to 50°C

HUMIDITY:

Up to 90% relative

POWER:

115 volts AC, single-phase $\pm 10\%$, 47 to
63 HZ
220 volts AC, single-phase $\pm 10\%$, 47 to
63 HZ
700 watts (4 K)

SYSTEM SECTION:

Processor and processor expansions
Data processing peripherals
Process input/output devices
Communications input/output and
terminals
Computer-to-computer couplers

MINI-CONTROLLERS

Mini-controllers are pre-engineered GA system interface units that enable multiple technologies to be combined in one computer subsystem. Their functional, plug-in design provides system modularity and interface versatility, eliminates redundant electronics, and permits quick, economical system configuring, expansion, and servicing. More than 70 off-the-shelf mini-controller units are

available for interfacing to and controlling instruments, sensors, mechanisms, devices, communication data sets, displays, keyboards, and a wide range of GA peripherals. By providing compatibility with existing customer device designs, design time and costs for implementation are substantially reduced. Field-proven integrated circuits throughout the advance-state design provide extremely high reliability. Mini-controllers can be housed in a 21-unit common enclosure with built-in power supply, cooling unit, and plug-in connectors.

ANALOG I/O

The Analog I/O System enables the SPC-16 to communicate with analog devices, and to perform both digital-to-analog and analog-to-digital conversions. This capability enables the SPC-16 to be readily interfaced to such devices as thermocouples, strain gauges, pressure transducers, potentiometers and other analog input devices. It will also drive system control functions, recording devices and other systems elements requiring analog outputs.

1441 ANALOG I/O UNIT (AIO)

The 1441 Analog I/O performs both analog-to-digital and digital-to-analog conversions. It is used with the 1451, 1452 and 1453 Analog Input Multiplexers and the 1481 Analog Output Holding Amplifier to increase the system's capabilities.

1451 HIGH LEVEL SINGLE-ENDED INPUT MULTIPLEXER (SEM)

The 1451 can interface sixteen channels of bipolar ± 5 volt analog input with the 1441 analog I/O unit. It provides multiplexing input capability for single-ended analog signals and is suited for applications where ground potentials are small with respect to the signal voltages.

1452 HIGH-LEVEL DIFFERENTIAL INPUT MULTIPLEXER (HLM)

The 1452 provides analog input capability for 8 channels to the 1441 Analog I/O unit. Input signal range is ± 5 volts for use in applications where a significant common mode voltage is present.

1453 LOW-LEVEL DIFFERENTIAL INPUT MULTIPLEXER (LLM)

The 1453 provides analog input capability for 8 channels to the 1441 Analog I/O unit. Input signal range is ± 200 millivolts for use in applications where a significant common mode voltage is present.

1481 ANALOG OUTPUT HOLDING AMPLIFIER (OHA)

Each 1481 provides 8 channels of analog output at a range of ± 5 volts. Each output channel maintains the voltage to within 5 mv per second and if fully buffered with low output impedance to make its output voltage almost independent of its load.

ANALOG SIGNAL CONDITIONERS

Used when standard output ranges must be modified to meet specific external circuit requirements.

ANALOG OUTPUT TERMINATION MODULES

Provides screw-down terminals or rugged multi-point cable connectors for terminating field wiring to customer equipment. Can be mounted to customer housing or in GA 1900 racks and enclosures.

DIGITAL INPUT/OUTPUT

The Digital Input Subsystem permits computer-directed real-time acquisition and transmission of digital equipment information in 16-bit parallel groups. The data word may consist of fields of binary-coded-decimals, binary discrete bits, or any other pattern.

1411 DIGITAL DIFFERENTIAL INPUT UNIT (DDI)

Provides a digital input signal interface to the processor for high speed applications or for applications requiring high-level electrical isolation. Can be used as an input unit for pulse type or dc signals, as an interface with remote equipment having low-power logic signal outputs, or as a line receiver. Each DDI module has 16 digital input circuits.

1431 DIGITAL INPUT RELAY RECEIVER (IRR)

The 1431 provides a digital input signal interface to the processor for applications that require electrical isolation. Electrical isolation of processor input signals is provided through the use of relay coil input circuits. The low power requirements of the input relay coils make the unit suitable in applications having either hard contact of electronic signal sources. Each IRR has 16 digital input circuits.

1413 GENERALIZED INPUT BUFFER (GIB)

General-purpose input device used for parallel data communication between equipment using TTL-compatible circuitry. By wiring at the card connectors, GIB functions include input data buffers, data buffer flags,

strobes, line driver and card addressing. Can be used as a peripheral device input controller with paper tape readers, card readers, and other medium-speed peripherals.

1412 BUFFERED BIPOLAR POWER DRIVER (BPD)

Provides the interface between the processor and peripheral of control equipment requiring high-speed driver signals from the processor. Suited for applications that require electrical isolation with nominal common mode voltage rejection and for use as a relay driver, lamp driver, or line driver. Each BPD has 18 output drivers.

1432 BUFFERED CONTACT OUTPUT (BCO)

Provides the interface between the processor and peripheral of control equipment requiring electrically isolated drive signals from the processor. For applications requiring high common mode voltage capabilities with moderate operation time and for controlling electromechanical equipment, lamps, annunciators, or in driving long lines. Each DCO has 18 relay contact outputs.

1414 GENERALIZED OUTPUT BUFFER (GOB)

General-purpose output device used for parallel data communications between equipment using TTL-compatible circuitry. By wiring the card connectors, GOB functions include data buffers, data buffer flags, strobes, line receivers, and card addressing. Can be used as a peripheral controller with devices as printers, card punches, paper tape punches, etc.

DIGITAL SIGNAL CONDITIONING

Used when the standard input ranges must be modified to meet specific external circuit requirements.

DIGITAL TERMINATION

Provides screw-down terminals or rugged multi-point cable connectors for terminating field wiring to customer equipment. Can be mounted in customer housing or in GA 1900 racks and enclosures.

COMMUNICATIONS I/O

GA communications I/O units interface the computer to data transmission devices. These units are available for synchronous or asynchronous modes of operation, half or full duplex, in a wide variety of data rates.

1551 BELL SYSTEM 103 DATA SET CONTROLLER (DS103)

Provides a buffered interface between the processor and a Bell System Model 103 Data Set. Includes a serial-to-parallel/parallel-to-serial converter and a Bell System Interface. 110, 150 or 300 baud operation.

1541 BELL SYSTEM 103/801 DATA SET AND AUTOMATIC CALLING UNIT (DS103/801)

Provides a buffered interface between the processor and Bell System Model 103 Data Set with Model 801 Data Auxiliary Set (Automatic Calling Unit). Includes a serial-to-parallel/parallel-to-serial converter and a Bell System Interface. 110, 150 or 300 baud operation.

1552 BELL SYSTEM 202 DATA SET CONTROLLER (DS202)

Provides a buffered interface between the processor and a Bell System Model 202 Data Set. Includes a serial-to-parallel/parallel-to-serial converter and a Bell System Interface. 1050, 1200, or 1800 baud operation.

1542 BELL SYSTEM 202/801 DATA SET AND AUTOMATIC CALLING UNIT CONTROLLER (DS202/801)

Provides a buffered interface between the processor and a Bell System Model 202 Data Set with Model 801 Data Auxiliary Set (Automatic Calling Unit). Includes a serial-to-parallel/parallel-to-serial converter and Bell System Interface. Full duplex 1050, 1200, or 1800 baud operation.

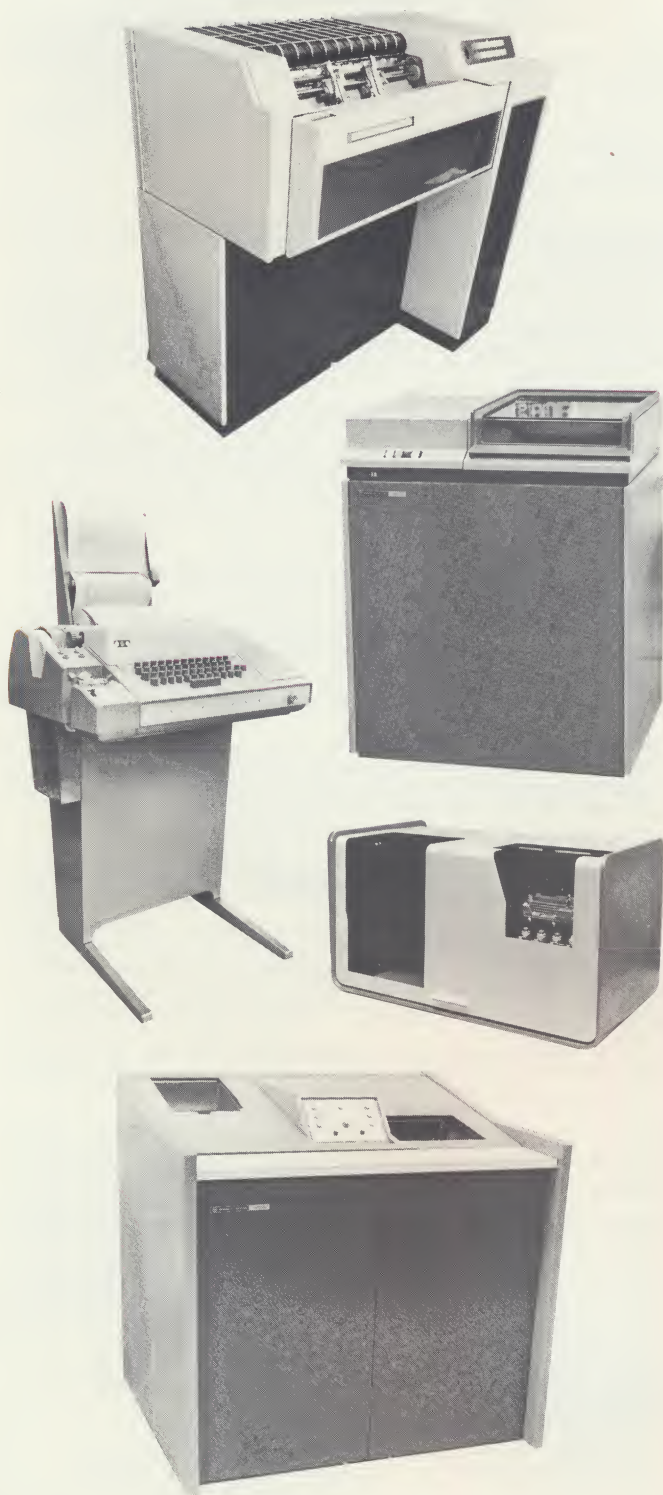
1553 BELL SYSTEM 201 DATA SET CONTROLLER (DS201)

Synchronous operation. Includes a serial-to-parallel/parallel-to-serial converter and Bell System Interface. Operates up to 2400 baud, speed determined by modem clock.

1554 BELL SYSTEM 201 DATA SET ADAPTER (DB201)

Provides for the connection to Data Sets with synchronous transmitting capabilities of up to 9600 bps. Controller is double buffered. Includes hardware sync character detection and Bell Model 201 interface.

DATA PROCESSING PERIPHERAL UNITS



1311 Card Reader: Serial reading of standard 80-column punched cards, 300 cards per minute.

1312 Card Reader/Punch: Serial reading and punching of standard 80-column punched cards. Reading rate, 500 cpm; punching rate, 300 cpm.

1321 Paper Tape Reader: Reading of 1-inch, 8-channel paper tape, 300 cps. Data is read into the processor as a direct image of the holes in the tape. Data encoding/decoding is a function of the operating program.

1322 Paper Tape Punch: Punching of 1-inch, 8-channel paper tape, 60 cps. Data characters are punched as a direct image of the processor data output. Data encoding/decoding is a function of the operating program.

1331 Magnetic Tape Unit: Magnetic tape input/output, 19.2 kilo-bytes per second. Data recording and retrieval format, 800 bpi, 9-track, odd parity. Handles reels with up to 2400 feet of storage.

1332 Magnetic Tape Unit: Magnetic tape input/output, 60 kilo-bytes per second (kbps). Data recording and retrieval format, 800 bpi, 9-track, odd parity. Handles reels with up to 2400 feet of storage.

1341 Disk Storage: Random-access memory; storage in excess of 5 million bytes. Storage media, interchangeable disk pack consisting of 6 disks (IBM 1361 or equivalent). Transfer rate, 156 kbps.

1342 Rotating Memory Storage: Fixed head; storage capacity, 32 kilo-bytes (kb) minimum, expandable in 32 kb increments to 256 kb maximum. Average access time to any byte in memory, 8.7 milliseconds.

1343 Rotating Memory Storage: Fixed head; storage capacity, 512 kb minimum, expandable in 32 kb increments to 1.2 million bytes maximum. Average access time to any byte in memory, 8.7 milliseconds.

1351 Printer: Printed output, 25 cps; line length, up to 132 columns per line (cpl). All 64 characters of the EBCDIC code are available.

1352 Printer: Printed output, 300 lines per minute; line length, up to 120 cpl (132 cpl, optional). All 64 characters of the EBCDIC code are available.

1361 Printer/Keyboard: Printed output, 15 characters per second (cps). Provides keyboard data entry into the processor.

1362 Printer/Keyboard (TTY): Punched paper tape or printed output, 10 cps. Provides 10 cps data entry into the processor via keyboard or paper tape reader.

1363 Printer/Keyboard (TTY): High-reliability version of the 1362, with identical operating characteristics and functions.

1364 Printer/Keyboard (TTY): High-reliability version of the 1362. Performs identical functions, but operates at 15 cps.

1371 Incremental Plotter: Recording of digital data in graphic form, 300 steps per second, 0.01 inches per step.

THE TOTAL-SOLUTION COMPANY

General Automation offers industry a unique, total-solution approach to automation projects. Through us, automation customers receive the best possible fourth-generation products in combination with extensive services in high-technology automation. GA users obtain from only one source all the hardware, software, and technological capabilities to successfully complete their automation project.

Computer-Based Automation Equipment

The GA Automation Products Division manufactures a comprehensive line of compatible, computer-based automation equipment. All GA products are designed around a modular-application theory, and may be used independently or as a total system, depending upon the size of the application. System interfacing, maintenance, and/or expansion is further simplified with wire-free, large-board construction and part interchangeability. Field expansion can be implemented without interrupting existing computerized operations. The GA line of supervisory systems, worker computers, and plug-in, pre-engineered mini-controllers excel in high performance-versus-cost, quality, reliability, compact packaging, and hardware/software compatibility.

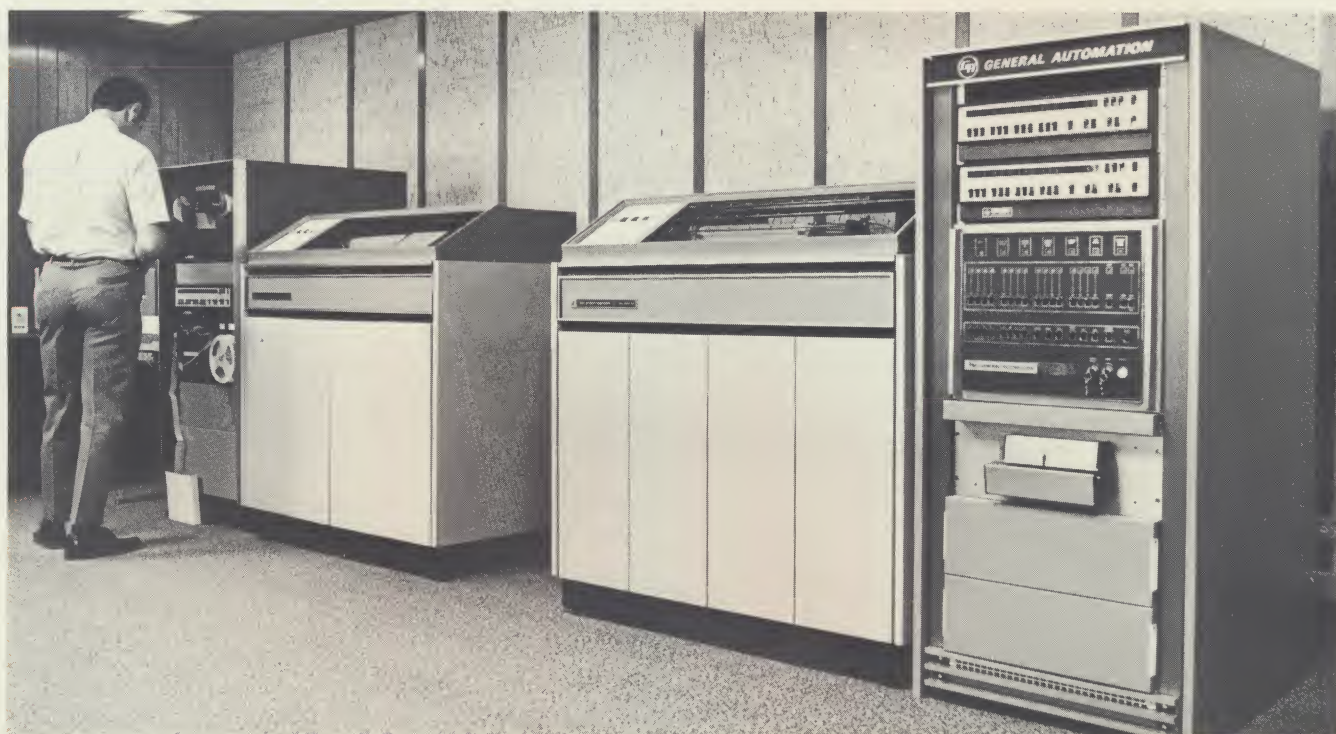
Automation Services

The GA Automation Sciences Division is the services counterpart to APD. ASD's Systems Analysts will provide a thorough analysis of any situation or condition for automation; its Programming and Systems Engineering

experts will implement a system to operate effectively at the lowest cost and shortest time possible. Higher productivity, increased profits, and fastest payback on minimum initial investment is thereby assured. These professionals offer the industrial automation market the highest computing skills available and the widest experience in applying these skills to automation projects. ASD supplies the entire scope of services required to implement any project, from in-depth feasibility and systems analyses through program development, systems engineering, and customer training, to full "turn-key" project responsibility. Organized on a modular basis, GA services may be obtained individually or in combination to fit each customer's requirements.

Sales and Field Service

GA users are strongly supported by a network of sales and service offices conveniently located throughout the U.S. and Europe. Each sales office is staffed with knowledgeable and highly experienced representatives. Backed up with the full resources of GA's automation expertise, these representatives can provide computer-based solutions to your automation project. In addition, Technical Application Centers (TAC) offer ASD service support on a convenient economical basis; each TAC contains a processing center, complete with software library. Installation and maintenance services are available from 110 branch service centers throughout the United States.



SALES AND SERVICE OFFICES

CALIFORNIA

1055 South East St.
ANAHEIM, 92805
(714) 778-4800
4600 El Camino Real
Suite 204
LOS ALTOS, 94022
(415) 941-5966

CONNECTICUT

111 Prospect St.
STAMFORD, 06902
(203) 325-3883

GEORGIA

2479 Peachtree Rd.
Suite 106
ATLANTA, 30309
(404) 261-6203

ILLINOIS

CALIFORNIA GA CORP.
2590 E. Devon Ave.
DES PLAINES, 60018
(312) 298-4650

MARYLAND

8555 16th St.
Suite 303
SILVER SPRING, 20910
(301) 587-7090

MASSACHUSETTS

235 Bear Hill Rd.
WALTHAM, 02154
(617) 899-6170

MICHIGAN

Industrial Bldg.
Suite 1900
232 West Grand River
DETROIT, 48226
(313) 965-9090

OHIO

9701 Brook Park Rd.
Suite 230
CLEVELAND, 44129
(216) 351-2275

PENNSYLVANIA

251 W. DeKalb Pike
Suite EG4
KING OF PRUSSIA, 19406
(215) 265-6525

TEXAS

Braniff Tower, Exchange Park
Suite 311
DALLAS, 75235
(214) 358-0271
6200 Hillcroft
Suite 505
HOUSTON, 77036
(713) 774-8716

INTERNATIONAL

BELGIUM

General Automation, Inc.
24 Blvd. de L'Empereur
BRUSSELS
Tel: 13-78-03

DENMARK

14 Holbergsgade
1057 COPENHAGEN
Tel: 11 24 48

ENGLAND

General Automation Ltd.
Wren House, Portsmouth Rd.
ESHER, Surrey
Tel: Esher 65764

FRANCE

General Automation, Inc.
33 Rue Galillee
No. 36, Iena Bldg.
PARIS 16
Tel: 553-47-79

GERMANY

ERA-Elektronische Rechenanlagen
51 AACHEN, Postfach 465
Werk: 5101 Verlauteheide
Lindenweg 23
Tel: RUF 02405/44-44
G.A. GmbH
2 HAMBURG 1
Gerhofstrasse 40
Tel: 04-11/35-19-21
G.A. GmbH
6200 WIESBADEN
Frauenlobstr. 9
Tel: 06121-83073



GENERAL AUTOMATION, INC. 1055 South East Street, Anaheim, California 92805, (714) 778-4800



BUSINESS REPLY MAIL

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

FIRST CLASS
Permit No. 423
Anaheim, Calif.

POSTAGE WILL BE PAID BY

General Automation, Inc.
1055 South East Street
Anaheim, California 92805

Thank you for your interest in General Automation

The information you requested is attached. For more information, complete and return the attached reply card. For instant action, call our nearest regional sales office (listed on the back).



GENERAL AUTOMATION, INC.

1055 South East Street
Anaheim, California 92805

Phone (714) 778-4800 TWX 910-591-1695

**FIELD
OFFICES**

8555 16th Street - Suite 303
Silver Spring, Maryland 20910
(301) 587-7090

111 Prospect Street
Stamford, Connecticut 06901
(203) 325-3883

2590 E. Devon Avenue
DesPlaines, Illinois 60018
(312) 298-4650

Industrial Bldg. - Suite 1900
232 West Grand River
Detroit, Michigan 48226
(313) 965-9090

9701 Brook Park Road - Suite 230
Cleveland, Ohio 44129
(216) 351-2275

251 W. DeKalb Pike - Suite EG4
King of Prussia, Pennsylvania 19406
(215) 265-6525

1055 South East Street
Anaheim, California 92805
(714) 778-4800
TWX 910-591-1695

4600 El Camino Real - Suite 204
Los Altos, California 94022
(415) 941-5966

6200 Hillcroft - Suite 505
Houston, Texas 77036
(713) 774-8716

**INTERNATIONAL
OFFICES**

GENERAL AUTOMATION, S.A., BRUSSELS
24 Boulevard de l'Empereur
1000 Brussels, Belgium
13 78 03

GENERAL AUTOMATION, LTD. - LONDON
c/o EAL, Victoria Road
Burgess Hill, Sussex, England
(4446) - 42400 & 42401

GENERAL AUTOMATION, S.A., PARIS
88 Avenue Victor Hugo
Paris 16e, 75 France
704-71-57

GENERAL AUTOMATION - DENMARK
14 Holbergsgade
1057 Copenhagen K, Denmark
11 24 48

GENERAL AUTOMATION - GERMANY
51 Aachen, Postfach 465
5101 Verlauteheide, Lindenweg 23,
Germany 2405 4444

GENERAL AUTOMATION - WIESBADEN
6200 Wiesbaden
Frauenlobstrasse 9, Germany
6121-83073

GENERAL AUTOMATION - HAMBURG
2000 Hamburg 1
Gerhofstrasse 40, Germany
411-35 19 21

GENERAL AUTOMATION, INC.

I am interested in further information describing:

- ☐ System 18/30 ☐ SPC-12 Minicomputer ☐ SPC-16 Minicomputer
☐ Interface Modules ☐ Application Engineering Software Assistance

My application is _____

My requirement is ☐ Immediate ☐ Future

Name _____ Title _____

Company _____ Phone _____

Address _____

City _____ State _____ Zip _____

IHSP-04-5-71

General Automation provides industry with a unique total-solution approach to automation projects. GA users obtain from one source all of the computer hardware, software, and technological capabilities necessary to successfully complete their automation project.

SALES & SERVICE OFFICES

CALIFORNIA

4600 El Camino Real
Suite 204

*LOS ALTOS, 94022
(415) 941-5966

1055 S. East St.
ANAHEIM, 92805

1402 E. Chestnut
SANTA ANA, 92701
(714) 835-4804

TEXAS

Braniff Tower
Exchange Park
Suite 311
DALLAS, 75235
(214) 358-0271

4660 Beechnut
Suite 202
*HOUSTON, 77035
(713) 774-8716

ILLINOIS

CALIFORNIA GA CORP.
2590 E. Devon Ave.
*DES PLAINES, 60018
(312) 298-4650

OHIO

9701 Brook Park Rd.
Suite 230
*CLEVELAND, 44129
(216) 351-2275

GEORGIA

2479 Peachtree Rd.
Suite 106
*ATLANTA, 30309
(404) 261-6203

PENNSYLVANIA

251 W. DeKalb-Pike
Suite EG4
*KING OF PRUSSIA, 19406
(215) 265-6525

MARYLAND

8555 16th St.
Suite 303
SILVER SPRING, 20910
(301) 587-7090

CONNECTICUT

111 Prospect St.
STAMFORD, 06902
(203) 325-3883

INTERNATIONAL

England

General Automation, Ltd.
Wren House
Portsmouth Rd.
ESHER, Surrey
Tel: Esher 65764

Belgium

General Automation, Inc.
24 Blvd. de L'Empereur
BRUSSELS, Belgium
Tel: 137803

Germany

ERA-Elektronische Rechenanlagen
Postfach 465
51 AACHEN
Werk: Verlautenheide
Lindenweg 23

*Indicates TAC capabilities

GLOSSARY OF COMMONLY USED COMPUTER TERMS



GENERAL AUTOMATION, INC.

1402 East Chestnut, Santa Ana, California 92701
(714) 835-4804 TWX (910) 595-1780



GENERAL AUTOMATION, INC.

A

access time: The time interval between the instant at which information is: 1. Called for from storage and the instant at which delivery is completed, i.e., the read time. 2. Ready for storage and the instant at which storage is completed, i.e., the write time.

accumulator: A part of the logical-arithmetic unit of a computer. It may be used for intermediate storage to form algebraic sums, or for other intermediate operations.

addend: The number or digital quantity to be added to another (*augend*) to produce a result (*sum*).

adder: A device capable of forming the sum of two or more digital quantities.

address: A label, name, or number that designates a register, a memory location, or a device.

address, absolute: An address that indicates the exact storage location where the referenced operand is to be found.

address, effective: A modified address. The address used in a particular execution of a computer instruction.

address, indirect: An address of a location that contains the address of the operand rather than the operand itself.

address modification: The process of changing the address part of a machine instruction by means of coded instruction. Methods are: 1. *Indexing*: Adding an index to the address. 2. *Derelativization*: Adding a base address to the address. 3. *Indirect addressing*: Using the intermediate computed address to obtain another address from memory.

address, relative: A label used to identify a word in a routine or subroutine with respect to its position in that routine or subroutine.

algorithm: A fixed step-by-step procedure for accomplishing a given result.

alphanumeric or alphameric: Characters which are either letters of the alphabet, numerals, or special symbols.

analog: The representation of numerical quantities by means of physical variables; i.e., translation, rotation, voltage, or resistance.

analog-to-digital (A/D) converter: A device that changes physical motion or electrical voltage into digital factors.

and: A logical operator which has the property such that if X and Y are two logic variables, then the function "X AND Y" is defined by the following table:

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

The AND operator is usually represented in electrical notation by a centered dot ".", and in FORTRAN programming notation by an asterisk "*" within a Boolean expression.

and-gate: A signal circuit with two or more input wires. The output wire gives a signal only if all input wires receive coincident signals.

arithmetic unit: That portion of the hardware of a computer where arithmetic and logical operations are performed.

array: A series of items arranged in a meaningful pattern.

ASCII: *American Standard Code for Information Interchange.*

This proposed standard defines the codes for a character set to be used for information interchange between equipments of different manufacturers and is the standard for digital communications over telephone lines.

assemble: To prepare with an assembler an object language program from a symbolic language program by substituting either procedure-operation codes of machine-operation codes for the symbolic-operation codes.

assembler: A computer program that operates on symbolic input to produce machine instructions. A processor program for a procedural-oriented or machine-oriented programming system.

assembly language: An operational language composed of brief expressions which is translated by an assembler into a machine language. The language result (*object*) from the assembler is a character-for-character translated version of the original.

augend: The number or digital quantity to which another (*addend*) is added to produce a result (*sum*).

augment: To increase a quantity or address in order to bring it to its full value.

automation: The technique of making a process or system automatic. Automatically controlled operation of an apparatus, process, or system, especially by electronic devices. In present-day terminology, usually used in relation to a system whereby the electronic device controlling an apparatus or process also is interfaced to and communicates with a computer.

B

base: A number base. A quantity used implicitly to define some system of representing numbers by positional notation. *Radix*.

Baudot code: The standard five-channel teletypewriter code consisting of a start impulse and five character impulses, all of equal length, and a stop impulse whose length is 1.42 times all of the start impulse.

binary: A numbering system based on 2's rather than 10's. Only the digits 0 and 1 are used when written.

bit: An abbreviation of *binary digit*.

blank: The character that results in memory when an input record, such as a card column without punches, is read. The character-code that will result in the printing of nothing in a given position. When associated with paper tape, *blank* refers to a section of tape that has only the sprocket hole punched.

block: A group of consecutive words or characters considered or transferred as a unit, particularly with reference to input and output. The term is used sometimes in connection with magnetic tape as a synonym for *record*, or to refer to grouped records on tape.

block diagram: A chart setting forth the particular sequence of operations to be performed for handling a particular application.

Boolean algebra: An algebra named for George Boole. This algebra is similar in form to ordinary algebra, but with

classes, propositions, yes/no criteria, etc., for variables rather than numeric quantities. It includes the operator's AND, OR, NOT, EXCEPT, IF, THEN.

bootstrap: A short sequence of instructions, which when entered into the computer's programmable memory will operate a device to load the programmable memory with a larger, more sophisticated program — usually a loader program.

branch (synonymous with jump): An instruction which when executed may cause the arithmetic and control unit to obtain the next instruction from some location other than the next sequential location. It is one of two types, conditional or unconditional.

breakpoint: A point in a program where it may be interrupted by external intervention.

buffer: A device which stores information temporarily during data transfers. Sometimes, a power amplifier.

bus: A channel along which data can be sent.

byte: A sequence of binary digits usually operated upon as a unit.

C

calling sequence: A basic set of instructions used to begin, initialize or transfer control to and return from a subroutine.

card image: A representation in storage of a punched card.

carry: A condition occurring during addition when the sum of two digits in the same column equals or exceeds the number base. The digit to be added to the next higher column. The process of forwarding the carry digit.

central processor: The portion of a computer that consists of the three main sections — arithmetic and control, input/output, and memory.

channel: A path along which signals can travel. That portion of a computer memory to which a particular output station has access.

character (synonymous with item): One of a set of elements which may be arranged in ordered groups to express information. Each character has two forms: 1. A man-intelligible form, the graphic, including the decimal digits 0-9, the letters A-Z, punctuation marks, and other formatting and control symbols. 2. A computer-intelligible form, the code, consisting of a group of binary digits (*bits*).

clear: To replace information in a storage unit by zero (or blank, in some machines).

coding: To prepare a set of computer instructions required to perform a given action or solve a given problem.

coding, absolute: Coding in the basic machine language.

coding, derelativized: *Absolute coding* with all relative addresses modified by addition of their instruction's base addresses.

coding, relocatable: *Absolute coding* containing relative addresses, which when derelativized, may be loaded into any portion of a computer's programmable memory and will execute the given action properly.

coding, symbolic: Coding in which the instructions are written in non-machine language.

compile: To prepare with a compiler an object language program from a symbolic language program by substituting machine-operation codes for the symbolic operation codes.

compiler: A processor program for a scientific procedural programming system.

compiler language: A computer language, more powerful than an assembly language, which instructs a compiler in translating a source language into a machine language. The machine-language result (*object*) from the compiler is a translated and expanded version of the original.

complement: A quantity expressed to the base n , by one of the following rules, which is frequently used to represent the negative of the given quantity: 1. Complement on n ; subtract each digit of the given quantity from $n - 1$, add unity to the least significant digit, and perform all resultant carries. For example, the twos' complement of binary 11010 is 00110; the tens' complement of decimal 456 is 544. 2. Complement on $n - 1$; subtract each digit of the given quantity from $n - 1$. For example, the ones' complement of binary 11010 is 00101; the nines' complement of decimal 456 is 543.

computer: A device capable of accepting information, applying prescribed processes to the information, and supplying the results of these processes.

computer, off-line: A computer not actively monitoring or controlling a process.

computer, off-line: A computer actively monitoring or controlling a process.

conditional jump: The *jump* is subject to the result of a comparison made during the program.

connector: A symbol used to indicate the interconnection of two points on a flow chart.

contents: The information in a storage location.

control, open-loop: An operation where the computer applies control action directly to the process without manual intervention.

control, open-loop: An operation where computer-evaluated control action is applied by an operator.

control unit: That portion of a computer which directs the automatic operation of the computer, interprets computer instructions, and initiates the proper signals to the other computer circuits to execute instructions.

core memory: A programmable, random-access memory consisting of many ferromagnetic cores strung on wires in matrix arrays.

counter: A device or location which can be set to an initial number and increased or decreased by an arbitrary number.

cycle: A sequence of operations that is repeated regularly. The time it takes for one such sequence to occur.

D

data: A collection of facts, numeric and alphabetical characters, etc., which is processed or produced by a computer.

data link: Equipment which permits the transmission of information in data format.

data processing: A procedure for collecting data and producing a specific result.

debugging: The process of determining the correctness of a computer routine, locating any errors, and correcting them. Also, the detection and correction of malfunctions in the computer itself.

debugging, on-line: The act of debugging a program while time-sharing its execution with an on-line process program.

diagnostic routine: A test program used to detect and identify hardware malfunctions in the computer and its associated I/O equipment.

digit: One of the n symbols of integral value ranging from 0 to $n - 1$ inclusive in a scale of numbering of base n , e.g., one of the ten decimal digits — 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

digital: A description of any data that is expressed in numerical format.

digital-to-analog (D/A) converter: A device which transforms digital data into analog data.

disk memory: A non-programmable, bulk-storage, random-access memory consisting of a magnetizable coating on one or both sides of a rotating thin circular plate.

documentation: The group of techniques necessarily used to organize, present, and communicate recorded specialized knowledge.

double precision: Data requiring two computer words to gain greater accuracy. Often called *double length*.

driver: A small program which controls peripheral devices and their interface with the Central Processor.

drum memory: A memory consisting of a magnetizable coating on outer surface of a rotating cylinder. Drums are typically non-programmable, bulk-storage, random-access memories.

dump: A small program that outputs the contents of memory onto hard copy which may be listings, tape, or punched cards.

duplex circuit: A telegraph circuit permitting simultaneous two-way operation.

duplex, full: Method of operation of a communication circuit where each end can simultaneously transmit and receive.

duplex, half: Permits one-direction, electrical communication between stations. Technical arrangements may permit operation in either direction but not simultaneously. Therefore, this term is qualified by one of the following suffixes: S/O for send only; R/O for receive only; S/R for send or receive.

E

engineering units: Units of measure as applied to a process variable, e.g., PSI, Degrees F., etc.

error: The difference in value between actual response and

desired response in the performance of a controlled machine, system, or process.

exclusive or: A logical operator, which has the property such that if S and Y are two logic variables, then the function " $X \text{ ERA } Y$ " is defined by the following table:

X	Y	X ERA Y
0	0	0
0	1	1
1	0	1
1	1	0

The ERA operator is usually represented in electrical notation by an encircled plus sign "+". There is no equivalent FORTRAN symbol.

executive control program: A main system program designed to establish priorities and to process and control other programs.

F

feedback: The signal or data fed back to a commanding unit from a controlled machine or process to denote its response to the command signal. The signal representing the difference between actual response and desired response that is used by the commanding unit to improve performance of the controlled machine or process.

field: A set of one or more adjacent columns on a punched card(s) or one or more bit positions in a complete word(s) consistently used to record similar information.

file: A collection of records. An organized collection of information directed toward some purpose. The records in a file may or may not be sequenced according to a key contained in each record.

file maintenance: The processing of a master file required to handle the non-periodic changes in it. For example, changes in number of dependents in a payroll file, the addition of new checking accounts in a bank.

filter: Device to suppress interference which would appear as noise.

fixed point: A notation or system of arithmetic in which all numeric quantities are expressed by a predetermined number of digits with the point implicitly located at some predetermined position; contrasted with *floating point*.

flag: A bit(s) used to store one bit of information. A flag has two stable states and is the software analogy of a flip-flop.

flip-flop: A bi-stable device. A device capable of assuming two stable states. A bi-stable device which may assume a given stable state depending upon the pulse history of one or more input points and having one or more output points. The device is capable of storing a bit of information; controlling gates; etc. A toggle.

floating point: A form of number representation in which quantities are represented by a bounded number (mantissa) and a scale factor (characteristic or exponent) consisting of a power of the number base, e.g., $127.6 = 0.1276 \times 10^3$ where the bounds are 0 and 1.

flow chart: A graphical representation of a sequence of operations, using symbols to represent the operations, such

as: Compute, substitute, compare, GO TO, IF, read, write, etc. Flow Charts of different levels of generality can be drawn for a given problem solution. Terms used are: System flow chart, program flow chart, coding level flow chart, etc.

format: The predetermined arrangement of characters, fields, lines, page numbers, punctuation marks, etc. Refers to input, output and files.

FORTRAN: *Formula Translator*. The language for a scientific procedural programming system.

FORTRAN compiler: A processor program for FORTRAN.

G

gap: An interval of space or time associated with an area of data processing activity (record) to indicate or signal the end of that record.

garbage: Unwanted and meaningless information in memory.

H

hard copy: Any form of computer-produced printed document. Also, sometimes punched cards or paper tape.

hardware: The mechanical, magnetic, electrical and electronic devices of which a computer is built.

head: A device, usually a small electromagnet on a storage medium such as magnetic tape or a magnetic drum, that reads, records, or erases information on that medium. The block assembly and perforating or reading fingers used for punching or reading holes in paper tape.

hexadecimal: A notation of numbers in the base 16.

Hollerith: A 12-bit code used for recording characters in punched paper cards.

housekeeping: Coding which reserves, restores, and clears memory areas.

I

index: An integer used to specify the location of information within a table or program.

index register: A memory device containing an index.

initialize: A program or hardware circuit which will return a program, a system, or a hardware device to an original state.

input: The data supplied to a computer for processing. The device employed to accomplish this transfer of data.

instruction: A set of bits which will cause a computer to perform certain prescribed operations. A computer instruction consists of: 1. An operation code which specified the operation(s) to be performed. 2. One or more operands (or addresses of operands in memory). 3. One or more modifiers (or addresses of modifiers) used to modify the operand or its address.

instruction register: A counter that indicates the instruction currently being executed.

Interface: A concept involving the specification of the interconnection between two equipments having different functions.

interrupt: A break in the normal flow of a system or program occurring in such a way that the flow can be resumed from that point at a later time. Interrupts are initiated by signals of two types: 1. Signals originating within the computer system to synchronize the operation of the computer system with the outside world (e.g., an operator or a physical process). 2. Signals originating exterior to the computer systems to synchronize the operation of the computer system with the outside world (e.g., an operator or a physical process).

item: See *character*, also *record*.

J

jump: See *branch*.

L

label: An ordered set of characters used to symbolically identify an instruction, a program, a quantity, or a data area.

language: A defined group of representative characters or symbols, combined with specific rules necessary for their interpretation. The rules enable an assembler or compiler to translate the characters into forms (such as digits) meaningful to a machine, a system, or a process.

left justified: A field of numbers (decimal, binary, etc.) which exists in a memory cell, location or register, possessing no zeros to its left. For example,

3	7	2		2	4	0
---	---	---	--	---	---	---

library programs: A software collection of standard routines and sub-routines by which problems and parts of problems may be solved on a given computer.

line printer: A printing device that can print an entire line of characters all at once.

linkage: A means of communicating information from one routine to another.

literal: A number or item of data to be reproduced character-for-character, whose symbolic value and actual value are identical. For example, a number stated as "270, a *literal*", would have a value of 270.

loader: A program that operates on input devices to transfer information from off-line memory to on-line memory.

location: A storage position in memory uniquely specified by an address.

log: A record of values and/or action for a given function.

loop: The repeated execution of a series of instructions for a fixed number of times.

M

machine language: A language written in a series of bits which are understandable by, and therefore instruct, a computer. The "first level" computer language, as compared to a "second level" assembly language or a "third level" compiler language.

macro: A source-language instruction from which many machine-language instructions can be generated (see *compiler language*).

magnetic core: An element for switching or storing a bit of information in a computer. Thousands of cores may be used in one computer.

magnetic core storage: The process of storing information on magnetic memory elements for later use by a computer.

magnetic disk storage: A storage device or system consisting of magnetically coated metal disks.

magnetic tape storage: A storage system that uses magnetic spots, representing bits, on coated plastic tape.

mask: A machine word that specifies which parts of another machine word are to be operated on.

memory: A device or media used to store information in a form that can be understood by the computer hardware.

memory, bulk: Any non-programmable large memory, i.e., drum, disk.

memory cycle time: The minimum time between two successive data accesses from a memory.

memory, random access: A memory whose information media are organized into discrete locations, sectors, etc., each uniquely identified by an address. Data may be obtained from such a memory by specifying the data address(es) to the memory, e.g., core, drum, disk, cards.

message: A group of words, variable in length, transporting an item of information.

microsecond: One-millionth of a second.

millisecond: One-thousandth of a second.

mnemonic code: Computer instructions written in a meaningful notation, e.g., LDB, STB.

modem: A contraction of *modulator-demodulator*. The term may be used with two different meanings: 1. The modulator and the demodulator of a modem are associated at the same end of a circuit. 2. The modulator and the demodulator of a modem are associated at the opposite ends of a circuit to form a channel.

monitor: An operating programming system which provides a uniform method for handling the real-time aspects of program timing, such as scheduling and basic input/output functions.

multiprogramming: A technique for handling numerous routines or programs seemingly simultaneously by overlapping or interleaving their execution, that is, by permitting more than one program to time-share machine components.

N

noise: An extraneous signal in an electrical circuit capable of interfering with the desired signal. Loosely, any disturbance tending to interfere with the normal operation of a device or system.

normalize: A computer operation that shifts all bits of a word to the right or left to accommodate the maximum number of bits.

not: A logic operator having property that if P is a logic quantity then the quantity "**NOT P**" assumes values as defined in the following table:

P	NOT P
0	1
1	0

The NOT operator is represented in electrical notation by an overline, e.g., $\overline{P}$ and in FORTRAN by a minus sign "-" in a Boolean expression.

numerical control: The control of machine tools by mechanical devices.

O

object program: The absolute coding output by a processor program. See *source program*.

octal: Pertaining to the number base eight. Pertaining to any set of exactly eight characteristics.

offset: The count value output from an A/D converter resulting from a zero input analog voltage. Used to correct subsequent non-zero measurements.

operand: Any quantities entering or arising in an operation. An operand may be an argument, a result, a parameter, or an indication of the location of the next instruction.

operating system: A group of programming systems operating under control of a data processing monitor program.

operation (OP) code: That part of an instruction designating the operation to be performed.

operation, parallel: Operates on all bits of a word simultaneously.

operation, serial: The flow of information through a computer in time sequence, usually by bit but sometimes by characters.

or: A logic operator having the property that if P and Q are logic quantities then the quantity "**P OR Q**" assumes values as defined by the following table:

P	Q	P OR Q
0	0	0
0	1	1
1	0	1
1	1	1

The OR operator is represented in both electrical and FORTRAN terminology by a "+", i.e., $P+Q$.

or gate: An electric circuit that implements the OR operator.

origin: In coding, the absolute memory address of the first location of a program or program segment.

output: Information transferred from the internal storage of a computer to output devices or external storage.

overflow: In an arithmetic operation, the generation of a quantity beyond the capacity of the register or location which is to receive the result. Over-capacity. The information contained in an item of information which is in excess of a given amount.

P

parallel operation: A type of information transfer performed by a digital computer in which all bits of a word are handled simultaneously.

parameter: In a subroutine, a quantity which may be given different values when the subroutine is used in different main routines or in different parts of one main routine, but which usually remains unchanged throughout any one such use. In a generator, a quantity used to specify input/output devices, to designate subroutines to be included, or otherwise to describe the desired routine to be generated.

parity bit: A binary digit appended to an array of bits to make the sum of all the bits always odd or always even.

parity check: A check that tests whether the number of ones (or zeros) in an array of binary digits is odd or even.

patch: A section of coding inserted into a routine to correct a mistake or alter the routine. Explicitly, transferring control from a routine to a section of coding and back again.

peripheral: Input/output equipment used to make hard copies or to read in data from hard copies (typer, punch, tape reader, line printer, etc.). Paper tape is considered hard copy for this definition.

point: A specific value or signal.

precision: The degree of discrimination with which a quantity is stated, e.g., a three-digit numeral discriminates among 1000 possibilities. *Precision* is contrasted with accuracy, i.e., a quantity expressed with 10 decimal digits of precision may only have one digit of accuracy.

priority: Order of importance.

procedural oriented language: A source language oriented to the description of procedural steps in machine computing, e.g., FORTRAN.

program: A plan for the solution of a problem. A complete program includes plans for the transcription of data, coding for the computer, and plans for the absorption of the results into the system. The list of coded instructions is called a routine. To plan a computation or process from the asking of a question to the delivery of the results, including the integration of the operation into an existing system. Thus programming consists of planning and coding, including numerical analysis, systems analysis, specification of printing formats, and any other functions necessary to the integration of a computer in a system.

pseudo (OP) instruction: A symbolic representation of information to a compiler or interpreter. A group of characters having the same general form as a computer instruction, but never executed by the computer as an actual instruction.

punched card: A piece of lightweight cardboard on which information is represented by holes punched in specific positions.

punched paper tape: A strip of paper on which characters are represented by combinations of holes.

Q

queue: Waiting lines resulting from temporary delays in providing service.

quotient: The quantity resulting from the division of a dividend by a divisor. If the dividend is not an even multiple of the divisor, the quantity left over is the remainder.

R

range: A characterization of a variable or function. All the values that a function may possess.

read: To copy, usually from one form of storage to another, particularly from external or secondary storage to internal storage. To sense the meaning of arrangements of hardware. To sense the presence of information on a recording medium.

reader: A device capable of sensing information stored in an off-line memory media (cards, paper tape, magnetic tape) and generating equivalent information in an on-line memory device (register, memory locations).

real time: Pertaining to the actual time during which a physical process transpires. Pertaining to the performance of a computation during the actual time that the related physical process transpires, in order that results of the computation can be used in guiding the physical process.

record: A collection of fields. The information relating to one area of activity in a data processing activity, i.e., all information on one inventory item. Sometimes called *item*.

register: A memory device capable of containing one or more computer bits or words. A register has zero memory latency time and negligible memory access time.

reset: To return a register or storage location to zero or to a specified initial condition.

right justified: A field of numbers (decimal, binary, etc.), which exists in a memory cell, location, or register, possessing no significant zeros to its right is considered to be right justified. 0 0 0 1 2 0 0 0 0 is considered to be a seven-digit field, right justified. 0 0 0 0 0 0 1 2 0 0 is a two-digit field, not right justified.

roundoff: To delete the least significant digit(s) of a numeral and to adjust the part retained in accordance with some rule.

routine: A series of computer instructions which performs a specific, limited task.

S

scale: To alter the units in which all variables are expressed to bring all magnitudes within bounds dictated by need, register size, or other arbitrary limits.

scale factor: One or more coefficients used to multiply or

divide quantities in a problem in order to convert them to a given magnitude, e.g., plus one to minus one.

Actual Number	Number in Storage	Scale Factor
9.0	.0009	4
.0009	9.0	-4

scan: To examine signals or data point-by-point in logical sequence.

scanner, analog input: A device which will, upon command, connect a specified sensor to measuring equipment and cause the generation of a digit count value which can be read by the computer.

segmented program: A program that has been divided into parts (segments) in such a manner that: 1. Each segment is self contained. 2. Interchange of information between segments is by means of data tables in known memory locations. 3. Each segment contains instructions to cause (or request a Monitor to cause) the transfer to the next segment.

sensor: A transducer or other device whose input is a quantitative measure of some external physical phenomenon and whose output can be read by a computer.

serial operation: A type of information transfer performed by a digital computer in which all bits of a word are handled sequentially.

setpoint: The required or ideal value of a controlled variable, usually pre-set in the computer or system controller by an operator.

shift: To move information serially right or left in a register(s) of a computer. Information shifted out of a register may be lost, or it may be re-entered at the other end of the register.

sign: The symbol or bit which distinguishes positive from negative numbers.

significant digit: A digit that contributes to the precision of a numeral. The number of significant digits is counted beginning with the digit contributing the most value, called the *most significant digit*, and ending with the one contributing the least value, called the *least significant digit*.

simulator: A device or computer program that performs simulation.

software: The programs or routines, and supporting documentation, which instruct the operations of a computer.

source language: The symbolic language comprised of statements and formulas used to specify computer processing. It is translated into object language by an assembler or compiler, and is more powerful than an assembly language in that it translates one statement into many items (see *macro*).

statement: In computer programming, a meaningful expression or generalized instruction in a source language.

subroutine: A series of computer instructions to perform a specific task for many other routines. It is distinguishable from a *main routine* in that it requires, as one of its parameters, a location specifying where to return to the main program after its function has been accomplished.

symbol table: A table of labels and their corresponding numeric values.

symbolic coding: Broadly, any coding system in which symbols other than actual machine operations and addresses are used.

syntax: The structure of expressions in a language. The rules governing the structure of a language.

system: A collection of parts or devices that forms and operates as an organized whole through some form of regulated interaction.

T

table: A collection of data, each item being uniquely identified either by some label or by its relative position.

table look-up: A procedure for obtaining the function value corresponding to an argument from a table of function values.

timeshare: To use a device for two or more interleaved purposes.

tracing routine or trace: A routine that provides a historical record of specified events in the execution of a program

track: The portion of a moving storage medium, such as a drum, tape, or disk, that is accessible to a given head position.

transducer: A device for converting energy from one form to another.

transfer vector: A transfer table used to communicate between two or more programs. The table is fixed in relationship with the program for which it is the transfer vector. The transfer vector provides communication linkage between that program and any remaining sub-programs.

trap: An unprogrammed conditional jump to a known location, automatically activated by hardware, with the location from which the jump occurred.

truncate: To terminate a computational process in accordance with some rule, e.g., to end the evaluation of a power series at a specified term.

V

variable: A quantity that can assume any of a given set of values.

W

word: A set of bits comprising the smallest addressable unit of information in a programmable memory.

word length: The number of bits in a word.

word time: See *memory cycle time*.

write: To deliver data to a medium such as storage.

Z

zero-suppression: The elimination of nonsignificant zeros in a numeral.